BTS Informatique de gestion 2e année
Option administrateur de réseaux locaux d'entreprise

Pacôme Massol – Thierry Savary

Architecture matérielle des systèmes informatiques

Cours spécifique ARLE

Directrice de publication : Valérie Brard-Trigo

Les cours du Cned sont strictement réservés à l'usage privé de leurs destinataires et ne sont pas destinés à une utilisation collective. Les personnes qui s'en serviraient pour d'autres usages, qui en feraient une reproduction intégrale ou partielle, une traduction sans le consentement du Cned, s'exposeraient à des poursuites judiciaires et aux sanctions pénales prévues par le Code de la propriété intellectuelle. Les reproductions par reprographie de livres et de périodiques protégés contenues dans cet ouvrage sont effectuées par le Cned avec l'autorisation du Centre français d'exploitation du droit de copie (20, rue des Grands Augustins, 75006 Paris).

Conseils généraux

Objectifs

Ce fascicule est le cours spécifique sur l'Architecture Matérielle des Systèmes Informatiques pour les étudiants ayant choisi l'option Administrateurs de Réseaux Locaux d'Entreprise.
Vous devez au préalable avoir étudié le 3988 (tomes 1 et 2). C'est particulièrement important : ce sont les notions fondamentales et il est probable qu'une part importante de l'examen porte sur le 3988. Si vous avez acquis ces notions vous pouvez alors très utilement compléter vos compétences avec l'étude de ce cours.
Bien sûr nul n'est devin sur ce qui vous sera demandé à l'examen. Les réseaux informatiques prennent une importance grandissante au sein des entreprises. Et les aspects de disponibilité et de sécurité sont des impératifs. L'évolution des difficultés des exercices de l'examen montre qu'on en demande de plus en plus aux étudiants : par exemple, ces dernières années, il a été demandé des exercices sur le routage bien sûr, mais aussi sur l'analyse de trames et de la translation d'adresses et de ports, des connaissances sur les réseaux sans fils, sur les VLAN et la voix sur IP… Bref vous devez, comme tout administrateur réseaux en entreprise, vous tenir informé des technologies, étudier leur théorie et leur pratique, en ne vous limitant pas à ce seul fascicule de cours.
Encore une fois il n'est pas question d'écrire une encyclopédie. Il serait anti-pédagogique de vous noyer dans un flux de détails techniques…
Rappelons que pour l'option ARLE, les questions d'architecture matérielle des systèmes informatiques représentent environ 10 points coefficient 5 à l'étude de cas, plus la soutenance de projet et l'épreuve pratique, cela fait coefficient 12 sur 22 !

Constitution du document

Ce fascicule se compose :

- de séquences de cours ;
- d'ateliers de travaux pratiques qui peuvent ne pas être en rapport direct avec la partie théorique, mais dont l'intérêt et la difficulté pédagogique les placent à cet endroit.

Indications d'équipement (matériel et logiciel)

À l'issue des ateliers du cours 3988, vous êtes l'heureux administrateur d'un petit réseau de deux postes : 1 sous windows XP, 1 sous Linux Debian, émulés par des machines virtuelles (Virtual PC2004).
Que vous faut-il de plus pour étudier ce cours ? Rien d'autre…
Est-il néanmoins intéressant, si financièrement je peux investir 200 à 300 € de plus, d'acheter autre chose ?
Oui bien sûr… Un peu plus bas vous trouverez une bibliographie indicative : les ouvrages réseaux sont nombreux, j'ai sélectionné ceux-là pour vous.

Conseils d'apprentissage

Tous les conseils d'apprentissage qui vous ont été prodigués dans votre scolarité comme dans les autres cours de ce cursus sont bien sûr valables.
Ce n'est pas la peine de dévorer entièrement ce cours en septembre : les épreuves de l'examen sont en mai-juin, vous risqueriez d'avoir oublié l'essentiel… Travaillez régulièrement…
Et, de temps en temps, n'hésitez pas à en « remettre une couche ». C'est-à-dire à revoir une séquence ou un chapitre, refaire un exercice ou un atelier, allez voir sur Internet si c'est expliqué pareil etc. N'hésitez pas non plus à demander à votre tuteur des exercices supplémentaires : il en a certainement une bonne collection et se fera un plaisir de vous les corriger.
Vous vous préparez à exercer un métier où le maître mot est « travail »… alors si nous commencions dès maintenant nos semaines de 60 heures ?
J'espère que vous aurez plaisir à étudier ce cours, merci et bon courage…

Bibliographie

Les ouvrages qui traitent des réseaux sont nombreux : certains sont généralistes (ils couvrent tous les aspects ou presque, comme ce cours par exemple), d'autres sont axés sur quelques notions. Voici une liste non exhaustive…
Avant d'acheter n'oubliez pas de vérifier la date de l'édition : les meilleurs de ces livres sont régulièrement remis à jour, un livre qui date de plus de 2 ans peut être considéré comme obsolète.
Pour un bon équilibre optez pour un livre généraliste réseau et un livre spécifique aux protocoles TCP/IP.

PETIT (BERTRAND), *Architecture des réseaux - Cours et exercices corrigés,* Ed. Ellipses
KARANJIT (S. SIYAN), *TCP / IP*, Ed. CampusPress, Coll. Student Edition
CASAD (JOE), *TCP / IP Notions fondamentales*, Ed. CampusPress

Ouvrages très intéressants mais qui commencent à dater :
DEAN (T.), *Réseaux informatiques 2e édition,* Reynald Goulet, (2002)
MONTAGNIER (J-L.), *Pratique des réseaux d'entreprise*, Ed. Eyrolles, (2004)
SIYAN (K.S.), *TCP/IP*, Ed. CampusPress, Coll. Macmillan, (1999)
JACQUENOD (F.), Administration des réseaux, Ed. CampusPress
KIRCH (O.) & DAWSON (T.), *Administration réseau sous Linux,* Ed. O'Reilly

Bien que nous ne souhaitions pas faire ici de publicité, il est incontournable de vous citer des ouvrages du fabricant Cisco :
CCNA INTRO, *Guide de certification*
Préparation à la certification

Liens

Les liens qui suivent sont ceux que je consulte régulièrement : il en existe d'autres. Lorsque des liens sont relatifs à une séquence en particulier, je ne les cite pas ici mais à la fin de la séquence. (liens testés et valides au 15 juillet 2007).

http://www.reseaucerta.org
http://adaig.free.fr/etudiant/site.php3
http://christian.caleca.free.fr
http://www.laissus.fr/cours/cours.html
http://www.urec.cnrs.fr/cours
http://abcdrfc.free.fr
http://www.themanualpage.org/reseau
http://www.docsdunet.com
http://www.reseaux-telecoms.net
http://www.labo-cisco.com

Et beaucoup d'autres : il est logique que le réseau Internet regorge de sources de formations et d'informations sur TCP/IP et les réseaux informatiques.
Vous avez de quoi vous occuper un bon bout de temps.

Sommaire

Séquence 1 : Trame Ethernet 7
Séquence 2 : Sur-réseau (supernetting) 13
Séquence 3 : Approfondissement DHCP 17
Atelier 1 : Chiffrement des communications 21
Atelier 2 : Mise en place d'un annuaire LDAP 37
Séquence 4 : Un peu de théorie sur LDAP 59
Atelier 3 : Gestion des disques 65
Atelier 4 : Gestion des pannes (reprise) 79
Atelier 5 : Les scripts shell 85
Atelier 6 : Le firewalling 99
Séquence 5 : Filtrage, NAT et PAT 113
Atelier 7 : Proxy 121
Atelier 8 : Le routage IP statique 131
Atelier 9 : Le routage IP dynamique OSPF 155
Atelier 10 : Audit de sécurité 173
Séquence 6 : NAS et SAN 181

Séquence 1

Trame Ethernet

Durée indicative : 1 heure

Cette séquence explique la composition d'une trame Ethernet.

▶ *Contenu*

1. Introduction 7

2. Les trames par la pratique 8

3. Format d'une trame Ethernet 9

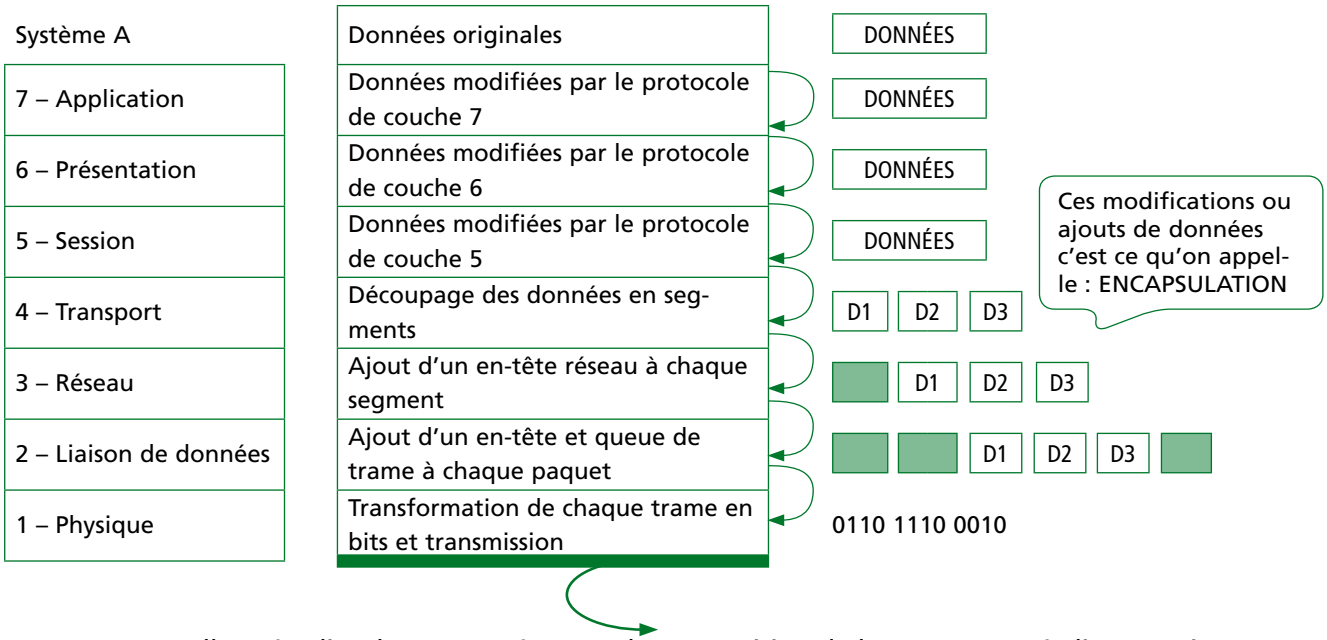
4. Formats de la partie données d'une trame Ethernet 10

▶ *Capacités attendues*

- **Savoir analyser une trame.**

1. Introduction

Vous vous rappelez le cheminement des données dans le modèle OSI ?

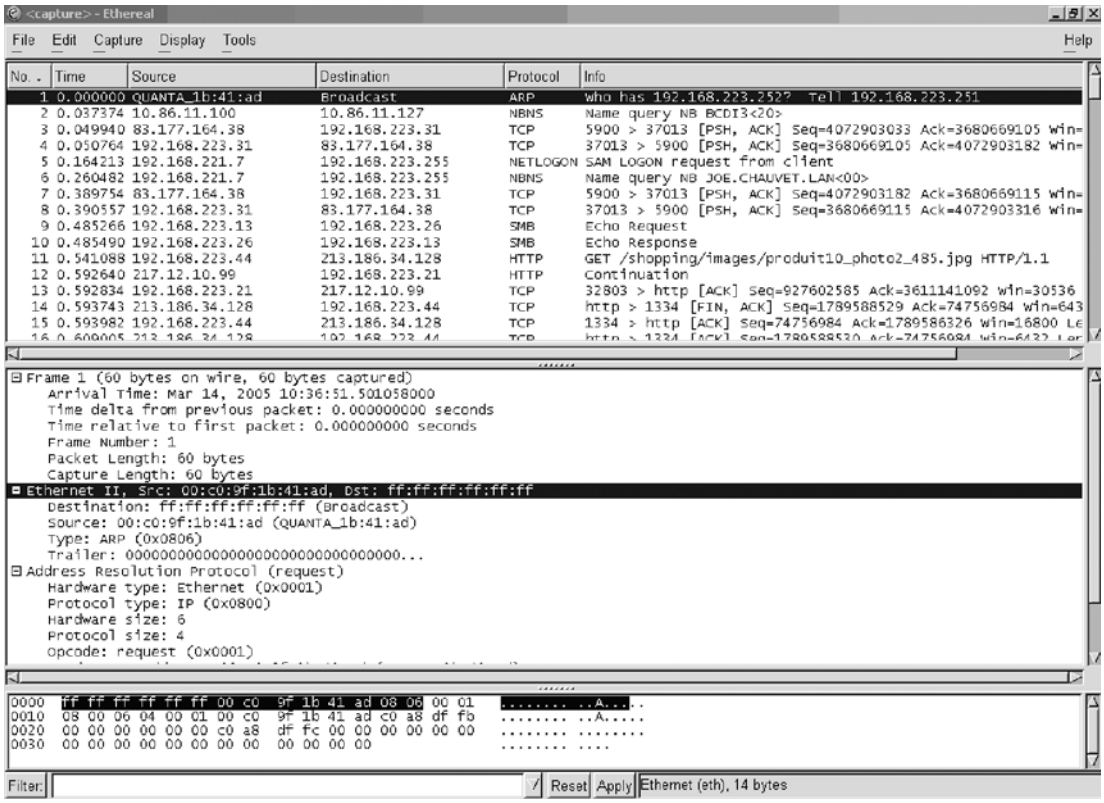

Système A
7 – Application
6 – Présentation
5 – Session
4 – Transport
3 – Réseau
2 – Liaison de données
1 – Physique
Données originales
Données modifiées par le protocole de couche 7
Données modifiées par le protocole de couche 6
Données modifiées par le protocole de couche 5
Découpage des données en segments
Ajout d'un en-tête réseau à chaque segment
Ajout d'un en-tête et queue de trame à chaque paquet
Transformation de chaque trame en bits et transmission
DONNÉES
DONNÉES
DONNÉES
DONNÉES
D1 D2 D3
D1 D2 D3
D1 D2 D3
0110 1110 0010
Ces modifications ou ajouts de données c'est ce qu'on appelle : ENCAPSULATION

Nous allons étudier dans cette séquence la composition de la trame, c'est-à-dire ce qui est « fabriqué » au niveau 2 et qui va ensuite être émis sur le support.

2. Les trames par la pratique

Nous allons tout naturellement commencer par capturer des trames et regarder à quoi cela ressemble. Il nous faut pour cela un outil logiciel de capture de trame (on parle aussi de snifer) : sans le doute le plus connu, et celui que j'ai utilisé est ethereal…
http://www.wireshark.org

Bien sûr ce fascicule n'a pas pour objet d'être un manuel d'utilisation de ce logiciel, il vous faudra donc faire cet apprentissage vous-même.
Commençons par une capture sans filtre et observons…



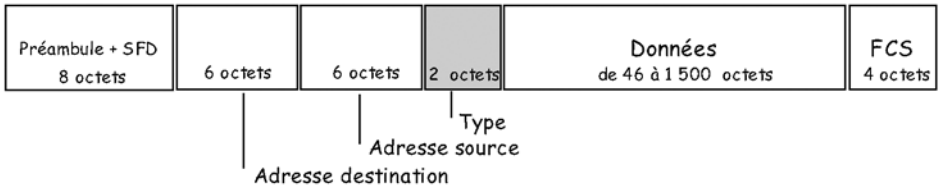
Le premier volet nous résume la liste des trames capturées.
Le deuxième volet analyse la trame mise en surbrillance.
Le troisième nous en donne le contenu en hexadécimal.

J'ai choisi ici une trame ARP pour en profiter pour faire un bref rappel : dans une trame nous devons avoir comme informations les @ IP et MAC de la source et du destinataire. Ici on voit que la machine d'@ IP 192.168.223.251 et d'@ MAC 00 : C0 : 9F : 1B : 41 : AD demande à la machine d'@ IP 192.168.223.252 quelle est son adresse MAC.
L'exercice le plus intéressant est concentré dans le troisième volet : comme le demandait l'étude de cas 2004, retrouver le sens des différents octets nécessite de comprendre comment une trame est constituée. Il n'est pas nécessaire de connaître par cœur ce qui suit, mais il faut comprendre et savoir faire. Le sujet 2004 rappelait d'ailleurs en annexe le schéma d'une trame et d'un datagramme…

3. Format d'une trame Ethernet

Voici le format d'une trame :

Format d'une trame Ethernet



SFD : *start frame delimiter indique le début de la trame*
FCS : *frame check sequence ou code de contrôle CRC*

- Le préambule sert à synchroniser la communication : 7 octets de valeur 1010 1010 + 1 octet SFD de valeur 1010 1011
- @ MAC de destination et source : chacune sur 6 octets
- Type de trame : 2 octets pour spécifier le protocole de niveau 3, exemple 08 00 pour IP, 08 06 pour ARP
- On trouve ensuite les données de 46 à 1500 octets : si le message fait moins de 46 octets on remplit avec des zéros, ça s'appelle la séquence de bourrage
- Enfin 4 octets pour le calcul d'intégrité de la trame

Reprenons notre capture ARP exemple, pour le volet 3 :

```
0000    ff  ff  ff  ff  ff  ff  00  c0    9f  1b  41  ad  08  06  00  01
0010    08  00  06  04  00  01  00  c0    9f  1b  41  ad  c0  a8  df  fb
0020    00  00  00  00  00  00  c0  a8    df  fc  00  00  00  00  00  00
0030    00  00  00  00  00  00  00  00    00  00  00  00
```

Alors analysons… mais nous avons besoin de deux informations importantes :

1. le logiciel Ethereal ôte le préambule de synchronisation même dans le 3e volet hexadécimal ;
2. les « 0000 » « 0010 » « 0020 » etc. représentent l'offset : l'adresse des octets à l'intérieur de la trame.

Alors comptons les octets comme de bons petits écoliers :

- 6 octets @ MAC destination = ff ff ff ff ff ff c'est une @ de broadcast destinée donc à toutes les machines du réseau.
- 6 octets @ MAC source = 00 c0 9f 1b 41 ad sous linux la commande ifconfig confirmerait que c'est bien l'@ MAC de ma carte.
- 2 octets type de trame = 08 06 cela confirme, ce code est celui du protocole ARP je comprends mieux pourquoi l'@ MAC de destination est un broadcast.
- La suite 46 octets ! les données terminées par une séquence de bourrage 00… pour faire les 46 octets minimum.

Que disent les données ? Allons-y doucement… À ce stade je vais juste vous demander de repérer la séquence : 00 c0 9f 1b 41 ad sur la ligne du dessous offset 0010 : tiens notre @ MAC source… c'est quoi les 4 octets c0 a8 df fb avant la séquence de bourrage 00 etc. ???

Allez je convertis ces 4 octets en décimal :

- c0 →192 – a8 →168 – df →223 – fb →251 ;
- regardez c0 a8 df fb c'est l'@IP de la machine qui demande la résolution d'adresse.

Bon, c'est clair. Obtenir l'@MAC dans une trame c'est facile car c'est au début, mais découvrir les @ IP… c'est dans les données qu'il faut chercher !

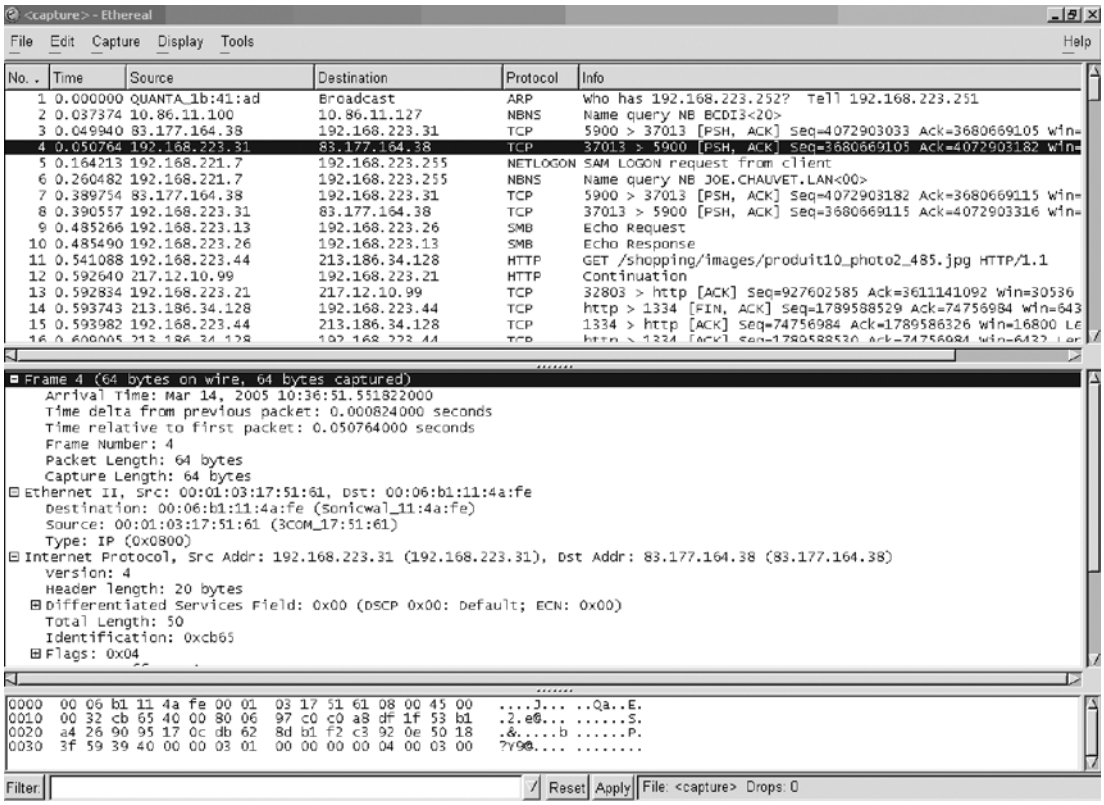
4. Formats de la partie données d'une trame Ethernet

Vous avez malheureusement bien lu : j'ai bien écrit formats avec un s !
Pourquoi ? Eh bien qu'est-ce qui se retrouve dans la partie données de la trame ?
Ce qui nous vient des couches supérieures bien sûr… Et à quoi ressemble ce qui vient du dessus ?
Tout dépend des protocoles utilisés !!! CQFD
Nous allons ici rester raisonnable et n'étudier que l'habituel datagramme IP :
Structure du datagramme IP (par lignes de 32 bits)

<table>
<tr><td colspan="4">1 4 5 8 9 16 17 19 20 24 25 32</td></tr>
<tr><td>Version</td><td>IHL</td><td>TOS (Type Of Service)</td><td>Longueur du datagramme - TL (Total Length)</td></tr>
<tr><td colspan="3">ID (IDentification)</td><td>FO Déplacement (Offset)</td></tr>
<tr><td>TTL (Time To Live)</td><td colspan="2">Protocole</td><td>Total de Contrôle (Header Checksum)</td></tr>
<tr><td colspan="4">Adresse IP Source</td></tr>
<tr><td colspan="4">Adresse IP Destination</td></tr>
<tr><td colspan="3">Options IP Éventuelles</td><td>Bourrage</td></tr>
<tr><td colspan="4">Données (de 2 à 65 517o)
…</td></tr>
</table>

IHL : *Internet Header Length ou longueur d'en-tête, en mots de 32 bits*
FO : *Fragment Offset indique si le fragment est suivi d'autres fragments*

Prenons un exemple, capturons



Le but du jeu est de trouver dans cette trame les @MAC et IP de la source et de la destination.
Alors reprenons la séquence hexadécimale du volet 3 :

```
0000    00 06 b1 11 4a fe 00 01    03 17 51 61 08 00 45 00
0010    00 32 cb 65 40 00 80 06    97 c0 c0 a8 df 1f 53 b1
0020    a4 26 90 95 17 0c db 62    8d b1 f2 c3 92 0e 50 18
0030    3f 59 39 40 00 00 03 01    00 00 00 00 04 00 03 00
```

Et commençons notre analyse :

- pas de préambule, donc on commence tout de suite par l'@MAC destination ;
- 00 06 b1 11 4a fe la voici notre adresse destination ;
- puis sur 6 octets : 00 01 03 17 51 61 voici notre @MAC source ;
- les 2 octets 08 00 c'est le type de la trame (ici une trame IP) ;
- bon attaquons-nous aux données… C'est une trame IP reprenons la structure du datagramme ;
- 1 octet 45 pour la version + IHL ;
- 1 octet 00 pour le TOS ;
- 2 octets 00 32 pour la longueur ;
- 2 octets cb 65 pour l'identification ;
- 2 octets 40 00 pour le FO et l'offset ;

- 1 octet 80 pour le TTL ;
- 1 octet 06 pour le protocole ;
- 2 octets 97 c0 pour le checksum ;
- ah enfin notre @IP source sur 4 octets : c0 a8 df 1f soit en décimal 192.168.223.31 ;
- et notre @IP destination sur 4 octets 53 b1 a4 26 soit en décimal 83.177.164.38.

Vérifions dans le volet 1 : gagné c'est bien les adresses IP affichées !

Voyez ce n'est pas très compliqué à partir du moment où on vous donne la structure et que vous comptiez méticuleusement les octets…

Pour la petite histoire regardez le volet 2 : ma 3com communique avec un adaptateur sonicwall, hi hi hi… c'est gentil monsieur le firewall : je lui écris et je vous ai caché dans la capture qu'il m'a gentiment répondu vous savez quoi ? son IP et son port d'administration et tant qu'à faire le mot de passe !!! Ah la sécurité n'est plus ce qu'elle était…

Séquence 2

Sur-réseau (supernetting)

Durée indicative : 1 heure

Cette séquence explique pourquoi et comment nous pouvons faire du sur-réseaux.

▶ *Prérequis*

- **L'adressage IP (voir cours 3988) et en particulier la notion de sous-réseaux doit être impérativement acquise sous peine de tout mélanger !**

▶ *Contenu*

1. Introduction 13

2. Comment ça marche ? 14

3. Mais alors… 15

▶ *Capacités attendues*

- **Être capable de calculer des sur-réseaux.**

1. Introduction

Le concept de sous-réseaux a été conçu en 1985 pour optimiser l'utilisation de l'espace d'adressage IP.

Rappelez-vous : chaque fois qu'une entreprise souhaite s'ouvrir sur l'Internet elle doit en particulier acquérir une adresse publique. Mais aujourd'hui, les adresses de classe A et B viennent à manquer et le NIC hésite donc fortement à en assigner. Problème ! Que peut-on faire ? Passer à IPv6 ou…
« J'ai demandé une @ de classe B : refusé on m'a dit, une classe C ou rien ! », mais avec une @ de classe C et un masque de sous-réseau classique on ne peut adresser que 254 hôtes.
Le principe de sur-réseau sera valable pour toutes les classes, mais il est plus particulièrement destiné aux réseaux de classe C : il s'agit de « travailler » le netmask pour augmenter la capacité du nombre d'hôtes.

Principe

La mise en sur-réseau, ou adressage en sur-réseau, consiste à **affecter un bloc d'adresses** de classe C par exemple plutôt qu'une seule adresse de classe B. Ce bloc, c'est-à-dire plusieurs adresses réseaux contiguës, peut alors être considéré comme une classe virtuelle située entre la classe B et la classe C.

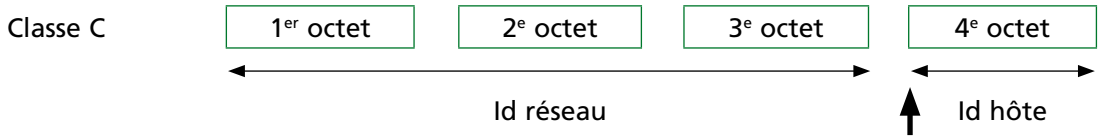
Cas d'utilisation

L'adressage en sur-réseau est conçu pour les fournisseurs d'accès Internet (FAI).
Pourquoi ?
Le NIC n'assigne de grands blocs d'adresses de classe C qu'aux providers (ISP), lesquels peuvent ensuite les redistribuer en petits blocs à des organismes qui ont besoin de plusieurs adresses, comme les FAI.

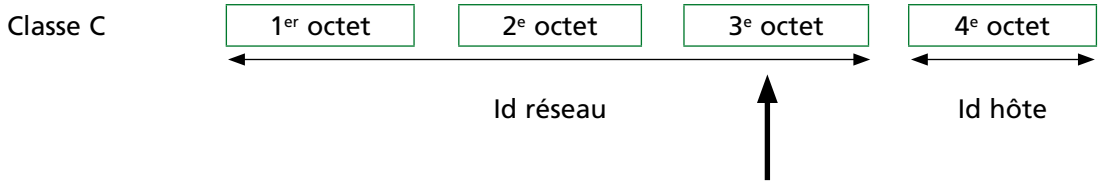
2. Comment ça marche ?

Facile : c'est le système inverse du sous-réseau.

Rappel des sous-réseaux : on prend quelques bits de l'id hôte pour obtenir plusieurs sous-réseaux, illustration…



Alors si on fait l'inverse :



On prend quelques bits de l'id réseau pour augmenter le nombre d'hôtes = on agrège plusieurs adresses de classe C, un bloc !

Exemple.
Soit l'adresse réseau 200.1.160.0 avec un masque 255.255.255.0
Les adresses hôtes vont donc de 200.1.160.1 à 200.1.160.254.

Voyons ça en binaire ; **IP toujours en binaire sous peine d'erreurs !**
@ réseau 1100 1000 . 0000 0001 . 1010 0000 . 0000 0000
masque sous-réseau 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000

Voyons ce que ça donnerait avec un autre masque :
@réseau 1100 1000 . 0000 0001 . 1010 0000 . 0000 0000
masque sous-réseau 1111 1111 . 1111 1111 . 1110 0000 . 0000 0000

ici nous avons pris 5 bits dans l'id réseau

Nous avons créé un espace d'adressage allant de :
+ petite @ en binaire = 1100 1000 . 0000 0001 . 1010 0000 . 0000 0001
+ petite @ en décimal = 200 . 1 . 160 . 1
+ grande @ en binaire = 1100 1000 . 0000 0001 . 1011 1111 . 1111 1110
+ grande @ en décimal = 200 . 1 . 191 . 254

CQFD (n'oubliez jamais l'explication binaire dans vos copies !)

3. Mais alors…

Revenons à l'intérêt d'un tel système : pour avoir un accès permanent à Internet vous savez qu'il faut demander une adresse réseau à un organisme chargé de la gestion des @ IP (NIC France par exemple). Vous vous rappelez que les adresses IPv4 commencent à s'épuiser (c'est pour cela qu'IPv6 est en gestation). Alors en attendant que tous les routeurs de la planète soient compatibles avec IPv6 on fait quoi ? Et bien lorsqu'une société, généralement ISP ou FAI, demande une adresse de classe A ou B à l'organisme, il est possible qu'elle lui soit refusée parce qu'il n'y en a plus ou presque plus. On lui affecte alors un bloc d'adresses de classe C (de 200.1.160.0 à 200.1.191.0 avec un netmask de 255.255.224.0 dans notre exemple) et le tour est joué !

Des entreprises peuvent également opter pour un adressage privé en sur-réseau. Pour la même raison que les ISP ou FAI une entreprise peut faire une demande d'adresse publique et se voit refuser une adresse de classe A ou B malgré un nombre important d'hôtes (elle ne s'appelle pas Microsoft par exemple). Il ne lui reste plus qu'à faire du sur-réseau !

Inconvénient pour les routeurs

Dans notre exemple de bloc de 200.1.160.0 à 200.1.191.0 avec pour netmask[1] 255.255.224.0 combien faut-il de routes dans la table de routage ?

CIDR = Classless Internet Domain Routing

La technique du CIDR permet de n'avoir qu'une seule entrée dans la table de routage.
Cette entrée est : + petite adresse du bloc, netmask.
Dans notre exemple cela donne : 200.1.160.0, 255.255.224.0
Une autre notation est utilisable (vous la connaissez déjà), laquelle ? **200.1.160.0 / 19**
(19 étant le nombre de bits consécutifs à 1 dans le netmask).
CIDR réduit donc la taille des tables de routage. Voir RFC 1519 pour plus de détails.

[1] *Deux fois que j'utilise le terme anglo-saxon netmask : sinon je devrai l'appeler masque de sous-réseau ! pour vous embrouiller sur les sur-réseaux y a pas mieux !*

Séquence 3

Approfondissement DHCP

Durée indicative : 2 heures

Cette séquence apporte quelques compléments au protocole DHCP.

▶ *Prérequis*

Avoir assimilé l'adressage IP (voir cours 3988).

▶ *Contenu*

1. Rappel 17

2. Comment ça marche ? 18

3. Routage et DHCP 19

4. DHCP et technologies sans fil 19

▶ *Capacités attendues*

Être capable d'expliquer le fonctionnement du service DHCP.

Être capable de résoudre les problèmes de routage lors de l'utilisation du service DHCP.

1. Rappel

Il est parfois préférable d'opter pour une configuration IP dynamique plutôt que statique :

- facilité d'attribution : configuration unique du serveur, pas des clients (qui peuvent être nombreux) ;
- évolution facilitée du plan d'adressage : seul le serveur est à reconfigurer ;
- facilité de gestion des postes itinérants : ordinateurs portables par exemple.

Pour cela on configure un serveur DHCP. DHCP (dynamic host configuration protocol) est un protocole qui peut fournir une configuration IP complète à un client : adresse IP, netmask, passerelle, adresses de serveurs WINS et DNS...

Cette configuration lui est attribuée à sa connexion sur le réseau pour un temps déterminé appelé le bail (eh oui comme la location d'un appartement !). Mais voyons ça d'un peu plus près…

2. Comment ça marche ?

Le client qui se « réveille » sur le réseau n'a pas de configuration IP, c'est entendu. Mais alors comment fait-il pour récupérer une configuration auprès d'un serveur : il doit envoyer une trame avec une adresse IP source ? Et il ne sait même pas l'adresse du ou des serveurs DHCP ?

❶ Le client émet une requête appelée DHCPDISCOVER :

- source : @IP 0.0.0.0 @MAC 00-04-75-D6-08-9D (celle-là il la connaît : c'est celle de sa carte réseau !) ;
- destinataire : @IP 255.255.255.255 @MAC FF-FF-FF-FF-FF-FF (broadcast : diffusion).

Traduction : « Y-a-t-il un serveur DHCP dans les parages pour me donner une configuration IP ? »

❷ Un serveur DHCP reçoit ce curieux message… regarde dans son étendue (plage d'adresses qu'il peut distribuer) ce qu'il peut faire et fait une proposition de configuration : requête DHCPOFFER :

- source : @IP du serveur @MAC carte du serveur ;
- destinataire : @IP 0.0.0.0 @MAC carte du client.

Traduction : « Je te propose la configuration suivante : @IP 192.168.1.66 netmask 255.255.255.0 passerelle 192.168.1.250 serveur wINS 192.168.1.3 et serveurs DNS 192.168.1.1 et 192.168.1.3, acceptes-tu ? »

❸ Un seul serveur DHCP = pas le choix mais même procédure que s'il y a plusieurs serveurs DHCP qui proposent. Le client répond à la 1re proposition par une requête DHCPREQUEST :

- source : @IP celle proposée @MAC carte du client ;
- destinataire : @IP 255.255.255.255 @MAC FF-FF-FF-FF-FF-FF.

Pourquoi une diffusion ?

Traduction de la requête client : « J'avertis tous ceux qui m'ont répondu que j'accepte la proposition de configuration IP du serveur Dalton ».

❹ Le serveur Dalton accuse réception par une requête DHCPACK : le client peut maintenant utiliser la configuration IP. Sur le serveur Dalton cette configuration, dans l'étendue, est maintenant cochée comme utilisée ; les éventuels autres serveurs ayant répondu à la demande notent que leur proposition n'a pas été retenue et qu'ils sont de nouveau libres.

Bail comme un appartement ? Quand cette durée est écoulée, que se passe-t-il ?

Par précaution, le « locataire » (le client), demande à son serveur DHCP de renouveler son bail à 50% du temps par une nouvelle requête DHCPREQUEST :

- réponse oui : requête DHCPACK = bail renouvelé pour la même durée ;
- pas de réponse ou réponse non : DHCPNACK, le bail continue (eh oui puisqu'il n'est pas terminé pas d'expulsion !).

Le client « locataire », anxieux, renouvelle cette demande à 80% du temps : requête DHCPREQUEST :

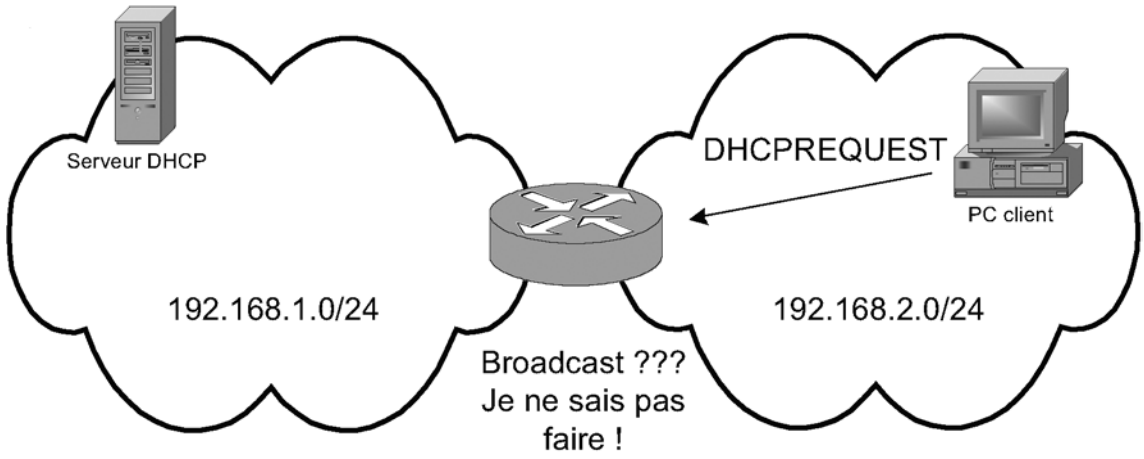
- réponse oui : requête DHCPACK = bail renouvelé pour la même durée ;
- pas de réponse ou réponse non : DHCPNACK, le bail continue (eh oui puisqu'il n'est pas terminé pas d'expulsion !).

Oui, mais quand le bail est fini ? Ou que le serveur répond DHCPNACK ?

Eh bien le client doit cesser d'utiliser sa configuration IP et en redemander une autre : retour à l'étape ❶.

3. Routage et DHCP

DHCP, comme tout protocole de la pile TCP/IP, est défini par les RFC 2131 et 2132.
Nous avons vu que la requête DHCPREQUEST de l'étape❶ est une diffusion… Problème : les routeurs ne routent pas les broadcast ! Donc, si le ou les serveurs DHCP ne sont pas sur le même réseau que le client, ça ne passe pas, illustration…



Solution ? Il faudrait installer un serveur DHCP sur le réseau du client !
Autre solution ? Oui la RFC 1542, si elle est prise en charge par le routeur permet de configurer le routeur en agent relais DHCP.
Agent relais DHCP = équipement capable de transmettre sur un autre réseau ou sous-réseau les différentes requêtes DHCP.
Cela peut être un routeur matériel ou un routeur logiciel bien sûr.

4. DHCP et technologies sans fil

Dans les technologies sans fil nous avons un point d'accès (access point). L'antenne pour simplifier…
Quelques modèles AP intègrent la RFC 1542 : il est possible d'y configurer un agent relais DHCP.
Mieux, quelques modèles intègrent un service DHCP : on peut alors directement configurer DHCP sur le AP !

Chiffrement des communications

Durée indicative : 2 heures

▶ *Objectif*

Mettre en place des connexions réseau chiffrées afin de rendre les données illisibles à qui n'est pas autorisé.

▶ *Conditions préalables*

Votre serveur web Apache doit être opérationnel.

▶ *Considérations techniques*

Nous poursuivons nos pérégrinations dans le logiciel Linux et entamons un point de plus en plus important dans le secteur informatique : le chiffrement des données. Nous mettrons en œuvre ce concept dans deux situations : chiffrement des données échangées par un serveur Web (vous avez déjà certainement utilisé, peut-être sans le savoir, le protocole *https*) puis chiffrement d'une communication via un terminal (un *telnet* chiffré SSH).

Introduction

Commençons par définir quelques termes de vocabulaire.

Qu'est-ce que chiffrer ?

C'est tout simplement l'opération par laquelle un message en clair est transformé en un message inintelligible pour tout intercepteur qui ne dispose pas de la **clé**.
La **clé** représente la méthode et/ou l'information nécessaire au déchiffrement.
Enfin, petite précision de vocabulaire : on devrait dire **chiffrement** et non pas cryptage, ainsi que **déchiffrement** et non pas décryptage. Le terme de décryptage consiste à tenter de découvrir le sens d'une information chiffrée sans y être autorisé.

Exercice 1

Vous êtes un chef gaulois et vous voulez envoyer un message stratégique à l'un de vos alliés, mais vous craignez que les Romains n'interceptent ce message. Dans ce cas, votre plan serait anéanti. Vous voulez donc que votre message, dans l'éventualité où il serait récupéré par un Romain soit incompréhensible.

1. Proposez une solution.
2. Donnez un exemple.

Vous avez compris que le but recherché en informatique est de rendre illisibles les données stockées ou échangées au travers des réseaux. Après les différents TP que nous avons réalisés, je pense que vous avez maintenant conscience que les réseaux ne sont pas sûrs car la plupart des informations circulent en « clair » (elles sont donc non chiffrées).

À propos de la clé

Si vous voulez que le ou les destinataires de votre message soient en mesure de le déchiffrer, il faut qu'il connaisse la clé.

Exercice 2

Imaginez plusieurs solutions pour fournir à vos interlocuteurs la clé pour déchiffrer vos messages. Réfléchissez aux risques et contraintes que cela suppose. Imaginez en particulier que vous n'avez pas un seul interlocuteur mais des dizaines (voire beaucoup plus !).

Vous comprenez où je veux en venir ? Rendre les messages illisibles est une première problématique. Mais transmettre et protéger la clé est sans doute une problématique encore plus complexe. Ça serait tellement plus simple si on pouvait chiffrer et déchiffrer, non pas avec la même clé, mais avec deux clés différentes. Avec une clé, connue de tous, je chiffrerais mon message. Dans ce cas, la diffusion de la clé ne poserait pas de problème. Ensuite, seule, la personne titulaire de la clé de déchiffrement (le destinataire du message) pourrait découvrir le contenu du message. Ça serait tellement bien…

Vous savez quoi ? Ça existe ! Ça s'appelle le **chiffrement à clés asymétriques** et c'est même la base de toutes les méthodes de chiffrement utilisées dans les réseaux informatiques. Tout d'abord, nous allons expliquer le principe puis nous le mettrons en œuvre dans deux situations.

Le chiffrement à clés asymétriques

Partons d'un petit exemple.

Supposons qu'une personne A (appelons-la Alice) désire transmettre des données chiffrées à une personne B (appelons-la Bernard). Bernard doit posséder un couple de clés :

- une clé privée **que lui seul possède et qui n'est jamais transmise** ;
- une clé publique que Alice doit aussi posséder. La clé doit lui être transmise ou bien elle peut se la procurer dans un annuaire.

Le processus de chiffrement sera le suivant :

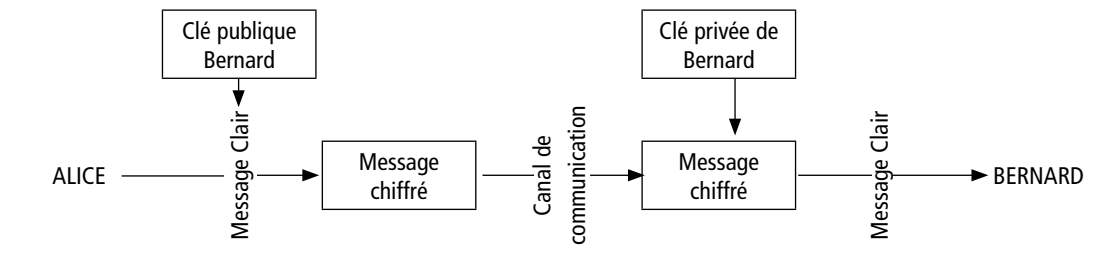


Figure 1 : exemple de chiffrement à clés asymétriques

Exercice 3

Je sais que ce schéma, qui paraît simple au premier abord, pose toujours des problèmes lorsque l'on interroge ensuite les étudiants. Voici quelques questions de « lecture » afin d'attirer votre attention sur les points importants :

- Qui est l'émetteur ? Qui est le destinataire ?
- Qui chiffre ? Avec quelle clé ? À qui appartient cette clé ?
- Qui déchiffre ? Avec quelle clé ? À qui appartient cette clé ?

Exercice 4

Alice souhaite maintenant envoyer un message chiffré à Charlotte. Comment doit-elle s'y prendre ?

Exercice 5

Sous quelle(s) condition(s), Bernard peut-il envoyer un message chiffré à Alice ?
En supposant que cette condition soit remplie et en vous inspirant de la figure 1, représentez l'échange d'un message chiffré de Bernard vers Alice.

SSL (secure socket layer)

Le protocole SSL *(secure sockets layer)* a été conçu par la société Netscape pour assurer une communication confidentielle et fiable entre deux applications (un client et un serveur), pour identifier le serveur et parfois le client. SSL nécessite un protocole de transport sûr comme TCP pour la transmission et la réception de données. Il permet la sécurisation de tout protocole applicatif s'appuyant sur TCP tels que HTTP, LDAP, NNTP, SMTP, FTP ou Telnet. Dans la pratique, les implémentations éprouvées de SSL s'appliquent surtout à HTTP (HTTPs) et LDAP❶ (LDAPs).
Le protocole est composé de deux couches qui s'insèrent entre le protocole de transport et les protocoles applicatifs :

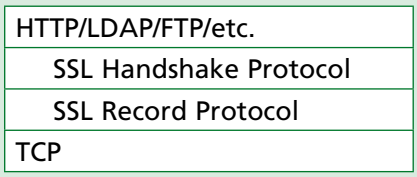


Figure 2 : le protocole SSL

Au niveau le plus bas, juste au-dessus d'un protocole de transport sûr, se trouve le *SSL Record Protocol* qui permet de transmettre de manière chiffrée les données. Celui-ci est utilisé pour encapsuler d'autres protocoles de plus haut niveau tel que le *SSL Handshake Protocol* qui permet au serveur et au client de s'authentifier et de négocier le fonctionnement de la session.

❶ *Nous traiterons de ce protocole dans la prochaine séquence.*

Quatre phases de négociation peuvent être distinguées.

Phase 1 : établissement des paramètres de sécurité

Cette phase a pour but d'établir le lien sécurisé. Le client envoie au serveur un message *client_hello* contenant des paramètres tels que, sa version de SSL utilisée, son identifiant de connexion, une liste d'algorithmes d'échange de clés et de chiffrement supportée par le client et classée selon la qualité de l'algorithme. Il envoie aussi un jeu de données qui sont générées aléatoirement. Après avoir envoyé ces requêtes, le client attend la réponse du serveur.

Phase 2 : authentification du serveur et échange des clés

Le serveur envoie *server_hello*, qui contient la version de SSL, les meilleurs algorithmes à utiliser, un jeu de données générées aléatoirement et le certificat[1] s'il existe. Dans le cas où le serveur n'a pas de certificat, ce premier message sera suivi d'un message *server_key_exchange* pour qu'il puisse tout de même transmettre sa clé publique. Ensuite, le serveur peut demander au client un certificat en envoyant un message *certificate_request*. Finalement, le serveur envoie le message *server_done*, qui signifie la fin de cette phase et que le serveur se met en attente.

Phase 3 : authentification du client et échange des clés

Le client doit vérifier que le certificat envoyé par le serveur est valide et que les autres paramètres sont corrects. Si le serveur a demandé au client d'envoyer un certificat, le client envoie un message *certificate* contenant le certificat (s'il n'a pas de certificat, il envoie un message *no_certificate*).

Il génère ensuite à partir de l'algorithme de chiffrement choisi une « pré » clé secrète *(pre_master_key)*. Il envoie un message *client_key_exchange* contenant la « pré » clé secrète chiffrée à l'aide de la clé publique du serveur. Pour finir cette phase, le client envoie un message *certificate_verify* pour indiquer que le certificat du serveur a été vérifié.

Phase 4 : fin

Le serveur vérifie éventuellement la validité du certificat du client. Il déchiffre ensuite la « pré » clé secrète à l'aide de sa clé privée. Le client et le serveur réalisent une même série d'opérations pour obtenir des clés secrètes de session à partir de la « pré » clé secrète et des données aléatoires échangées précédemment.

Le client envoie enfin le message *finished* qui valide l'échange de clés. Le serveur répond en envoyant son message *finished*. Les deux parties sont maintenant identifiées, le protocole *handshake* est terminé et les communications sécurisées (chiffrées avec la clé secrète générée) peuvent avoir lieu.
Toutes ces manipulations sont réalisées par le protocole mais il est bon que vous les connaissiez. Passons maintenant à l'administration de notre serveur Web.

[1] *Un certificat est une carte d'identité numérique identifiant une machine. Le certificat contient, entre autres, la clé publique. Nous reviendrons sur cette notion un peu plus loin.*

HTTPS : le web chiffré

Sur le serveur

Le traditionnel lien :
http://www.coagul.org/article.php3?id_article=351
Dans cette partie, je suppose que votre serveur web est en état de marche. Pour passer au web chiffré, il faut adjoindre à notre Apache un petit module nommé *mod_ssl*. (module qui fait partie du projet openssl), installons le paquet openssl :

```
mv2-debian:/# apt-get install openssl
[...] 0 mis à jour, 0 nouvellement installés, 0 à enlever [...]
```

Openssl est donc bien déjà installé. À partir de la version 2, le mod_ssl a été intégré à Apache : donc ce module a été installé en même temps que notre serveur, mais n'a pas été activé…

Vérifions notre version d'Apache (c'est juste pour s'échauffer).

```
mv2-debian:/# apache2 –v
Server version: Apache/2.2.3
Server built: Mar 27 2007 15:06:55
```

Vérifions mod_ssl avec Apache2 : est- il disponible ?

```
mv2-debian:/# ls /etc/apache2/mods-available
[...]
ssl.conf
ssl.load
[...]
```

mod_ssl est-il activé ?

```
mv2-debian:/# ls /etc/apache2/mods-enabled
[...] Pas de ssl en vue [...]
```

Alors activons-le : (la commande **a2enmod** est une contraction de apache2 enable module.)

```
mv2-debian:/# a2enmod ssl
Module ssl installed; run /etc/init.d/apache2 force-reload to enable
```

Bien sûr nous relancerons Apache2 le moment venu puisque nous modifions sa configuration. En attendant vérifions l'effet de notre commande :

```
mv2-debian:/# ls /etc/apache2/mods-enabled
[...]
ssl.conf
ssl.load
[...]
```

mod_ssl est maintenant actif.

Nous allons créer un certificat de sécurité pour notre service (ça crée donc nos clés publique et privée attachées à un certificat de sécurité) :

```
mv2-debian:/# apache2-ssl-certificate
-bash: apache2-ssl-certificate: command not found
```

(note : contrairement aux ateliers des fascicules 3988 destinés aux deux options, vous êtes ici dans la partie des administrateurs réseaux... c'est quoi la différence ? Un développeur d'applications sait débugger des messages d'erreur lors de la compilation de ses programmes mais n'aime pas les messages d'erreur lors d'installations ou de configurations de services réseaux, vous c'est l'inverse...)

Bon ici ce n'est pas très difficile à tirer la leçon suivante : lorsque 10 sources vous disent de taper la commande **xyz** et qu'il y a un tel message d'erreur... il faut trouver une explication :
http://www.phwinfo.com/forum/archive/index.php/t-178798.html
Cherchez... Vous trouverez et comprendrez qu'il s'agit d'un souci de version : ne cherchez pas non plus le paquet **apache2-common** il a été remplacé par **apache2.2-common** et aussi **apache2-utils**... qui sont déjà installés sur notre serveur ! Eh oui vous avez certainement trouvé que la commande **apache2-ssl-certificate** a disparue ! Et qu'il faut maintenant créer le certificat via les commandes de openssl :

http://www.free-4ever.net/index.php/Accueil et http://www.narno.com/blog/debian-apache2-ssl
Nous allons générer une clé privée pour le domaine **labocned.fr** nommée : **labocned.fr.key :**

```
mv2-debian:/etc/ssl# openssl genrsa –des3 –out labocned.fr.key 1024
Generating RSA private key, 1024 bit long modulus
...++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for labocned.fr.key: <tapez labocned par exemple>
Verifying – Enter pass phrase for labocned.fr.key: <re-saisissez le même>
```

Nous allons formuler une demande de certificat nommé **labocned.fr.csr**

```
mv2-debian:/etc/ssl# openssl req –new –key labocned.fr.key –out labocned.fr.csr
Enter pass phrase for labocned.fr.key: <saisissez votre pass: labocned ici>
You are about to be asked to enter information that will be incorporated
Into your certificat request.
What you are about to enter is what is called a Distinguished Name or DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

```
-----
Country Name (2 letter code) [AU]: <tapez fr>
Sate or Province Name (full name) [Some-State]: <laissez vide ENTER>
Locality Name (eg, city) []: <laissez vide ENTER>
Organization Name (eg, company) [Internet Widgits Pty Ltd]: <saisissez labocned>
Organizational Unit Name (eg, section) []: <saisissez labocned>
Common Name (eg, YOUR name) []: <laissez vide ENTER>
Email Address []: <laissez vide ENTER>
Please enter the following 'extra' attributes
To be sent with your certificate request
A challenge password []: <saisissez labocned par exemple, ce n'est pas raisonnable bien sûr, mais ici c'est un atelier...>
An optional company name []: <saisissez labocned>
```

Ouf !!!

Il faudrait faire signer ce certificat par un tiers de confiance... Bien. Nous allons créer un certificat de « test ». Pourquoi de « test » ? Et pourquoi pas un vrai de vrai ? Réfléchissons. Nous avons dit que le certificat était une sorte de pièce d'identité numérique pour un serveur. Imaginez que chacun d'entre nous se fasse sa propre carte d'identité ! Quelle valeur aurait-elle aux yeux des autres ? Pour qu'une carte d'identité soit prise au sérieux, il faut qu'elle soit produite par une autorité sérieuse. Dans la vraie vie, c'est la préfecture qui délivre les cartes d'identités. Dans le monde Internet, ce sont des **autorités de certification** qui s'en chargent (jetez par exemple un coup d'oeil sur http://www.certinomis.com).

En test nous allons le signer nous-même comme si nous étions autorité de certification :

```
mv2-debian:/etc/ssl# openssl genrsa –des3 –out ca.key 1024
mv2-debian:/etc/ssl# openssl req –new –x509 –days 365 –key ca.key –out ca.crt
```

À ce stade nous sommes positionnés dans **/etc/ssl** et regardons ce que nous avons :

```
mv2-debian:/etc/ssl# ls
ca.crt certs labocned.fr.key private
ca.key labocned.fr.csr openssl.cnf privkey.pem
```

Nous allons signer **labocned.fr.csr** avec **ca.crt**

```
mv2-debian:/etc/ssl# openssl x509 –req –in labocned.fr.csr –out labocned.fr.crt –CA ca.crt –CAkey ca.key –CAcreateserial –CAserial ca.srl
Signature ok
Subject=/C=fr/ST=Some-State/O=labocned/OU=labocned
Getting CA Private Key
Enter pass phrase for ca.key: <tapez votre pass, ici labocned>
```

À quoi ressemble un certificat ? Regardons **labocned.fr.crt**

```
mv2-debian:/etc/ssl# cat labocned.fr.crt
-----BEGIN CERTIFICATE-----
[...]
-----END CERTIFICATE-----
```

Lorsque l'on voudra interroger notre serveur Apache2 en mode sécurisé, celui-ci ira lire le fichier **ssl.conf** qui est maintenant dans **/etc/apache2/mods-enabled/**
Il cherchera dans ce fichier **ssl.conf** le path des certificats...
Aussi, nous allons copier les fichiers, **labocned.fr.key** et **labocned.fr.crt** dans un unique répertoire **ssl** que nous allons créer dans **/etc/apache2**

```
mv2-debian:/etc/apache2# mkdir ssl
mv2-debian:/etc/apache2# cp /etc/ssl/labocned.fr.csr /etc/apache2/ssl/labocned.fr.csr
mv2-debian:/etc/apache2# cp /etc/ssl/labocned.fr.key /etc/apache2/ssl/labocned.fr.key
```

Ajustons les permissions :

```
mv2-debian:/etc/apache2# chown –R root:root /etc/apache2/ssl/
mv2-debian:/etc/apache2# chmod –R o-rwx /etc/apache2/ssl/
```

Un serveur http sécurisé ne se positionne pas sur le port 80, mais sur le **port 443**. Il faut modifier **/etc/apache2/ports.conf** :

```
mv2-debian:/etc/apache2# vi /etc/apache2/ports.conf
Listen 80
Listen 443
```

Occupons-nous maintenant du site qui sera accessible en sécurisé. Gilbert et ses collègues garderont leur sites personnels respectifs bien sûr. Nous allons créer un nouveau site qui lui sera destiné à mettre à dispositision des données sécurisées. Créons-le à la racine du service dans **/var/www**

```
mv2-debian:/var/www# mkdir secure
```

Créons une page dans ce répertoire : **/var/www/secure/index.html**

```
<html>
  <head>
    <title>Page chiffrée</title>
  </head>
  <body>
    <h1>Bienvenue sur une page sécurisée</h1>
  </body>
</html>
```

Pour gérer tous ces sites, il nous faire appel à un **VirtualHost** (hôte virtuel). De retour dans **/etc/apache2/** nous pourrions déclarer **default-ssl** (inspiré de **default**) dans les sites disponibles, puis l'activer...

```
mv2-debian:/etc/apache2# cp /etc/apache2/sites-available/default /etc/apache2/sites-available/default-ssl
```

Configurons **/etc/apache2/sites-available/default-ssl** :

```
mv2-debian:/etc/apache2/sites-available# vi default-ssl
[etc comme ci-dessous] et activons le site...
```

Activons ce site :

```
mv2-debian:/etc/apache2# a2ensite default-ssl
```

Pour ce que nous avons à réaliser (un site sécurisé /var/www/secure et des sites personnels utilisateurs) nous pouvons directement éditer /etc/apache2/sites-enabled/000-default.

```
NameVirtualHost *:80
NameVirtualHost *:443
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
```

```
        Order allow,deny
        allow from all
    # This directive allows us to have apache2's default start page
    # in /apache2-default/, but still have / go to the right place
    # RedirectMatch ^/$ /apache2-default/
</Directory>

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options ExecCGI –MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
</Directory>

ErrorLog /var/log/apache2.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

CustomLog /var/log/apache2/access.log combined
ServerSignature on

Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
        Options Indexes MultiViews FolowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0::1/128
</Directory>
</VirtualHost>
<VirtualHost *:443>
        ServerAdmin webmaster@localhost
        ServerName secure.labocned.fr
```

```
        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/labocned.fr.crt
        SSLCertificateKeyFile /etc/apache2/ssl/labocned.fr.key

        ServerSignature EMail

        ErrorLog /var/log/apache2/error_secure.log
        CustomLog /var/log/apache2/access_secure.log combined

        DocumentRoot /var/www/secure/
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory "/var/www/secure/">
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Et rechargeons le service :

```
mv2-debian:/etc/apache2# /etc/init.d/apache2 reload
Forcing reload of web server (apache2)...Apache/2.2.3 (Pass Phrase
 Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server secure.labocned.fr:443 (RSA)
Enter pass phrase: <tapez votre phrase pass>
```

Vérifions que tout a été pris en compte. Contrairement au serveur web classique qui écoute sur le port TCP 80, notre serveur https écoute sur le port 443 :

```
# netstat -ltn
Connexions    Internet    actives (seulement serveurs)
Proto Recv-Q  Send-Q      Adresse locale      Adresse distante    Etat
[...]
tcp           0           0                   0.0.0.0:80          0.0.0.0:*           LISTEN
tcp           0           0                   0.0.0.0:443         0.0.0.0:*           LISTEN
[...]
```

Notez que le serveur web classique est toujours actif.

Sur le client

Vérifions dans les options Internet que notre navigateur supporte bien le SSL.

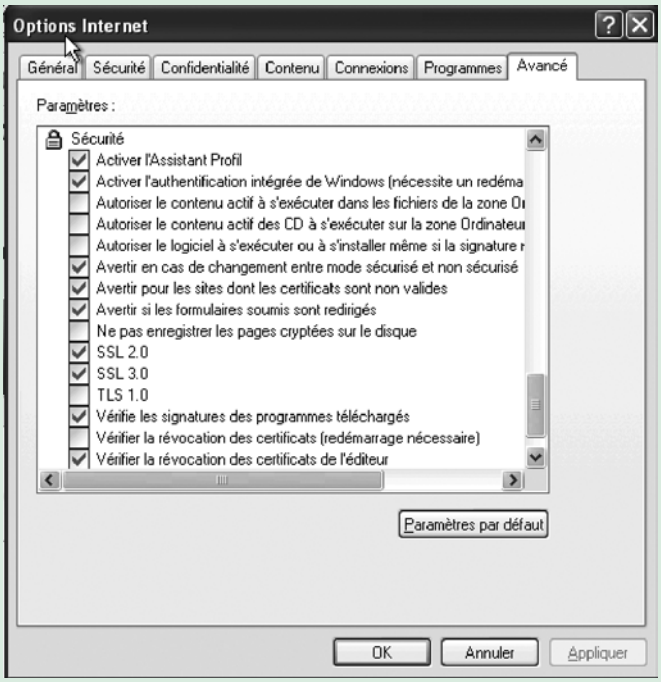

Options Internet
Général
Sécurité
Confidentialité
Contenu
Connexions
Programmes
Avancé
Paramètres :
Sécurité
Activer l'Assistant Profil
Activer l'authentification intégrée de Windows (nécessite un redéma
Autoriser le contenu actif à s'exécuter dans les fichiers de la zone Or
Autoriser le contenu actif des CD à s'exécuter sur la zone Ordinateur
Autoriser le logiciel à s'exécuter ou à s'installer même si la signature n
Avertir en cas de changement entre mode sécurisé et non sécurisé
Avertir pour les sites dont les certificats sont non valides
Avertir si les formulaires soumis sont redirigés
Ne pas enregistrer les pages cryptées sur le disque
SSL 2.0
SSL 3.0
TLS 1.0
Vérifie les signatures des programmes téléchargés
Vérifier la révocation des certificats (redémarrage nécessaire)
Vérifier la révocation des certificats de l'éditeur
Paramètres par défaut
OK
Annuler
Appliquer

Figure 3 : support de SSL

Vérifions également la puissance de chiffrement, c'est marqué dans la page « À propos de... » :

À propos de Internet Explorer
Microsoft
Internet Explorer
Version : 6.0.2900.2180.xpsp_sp2_rtm.040803-2158
Niveau de cryptage : 128-bit
ID de produit (Product ID) :76413-008-3845645-22536
Versions des mises à jour :; SP2;
Basé sur NCSA Mosaic. NCSA Mosaic(TM) a été développé par le National Center for Supercomputing Applications à l'Université de l'Illinois à Urbana-Champaign.
Copyright ©1995-2004 Microsoft Corp.
OK

Figure 4 : puissance de chiffrement de IE

La puissance de chiffrement supporte en général un chiffrement sur 128 bits.
Accédons au site sécurisé en indiquant l'URL et en spécifiant **https** car sinon on atterrit sur le site non chiffré :

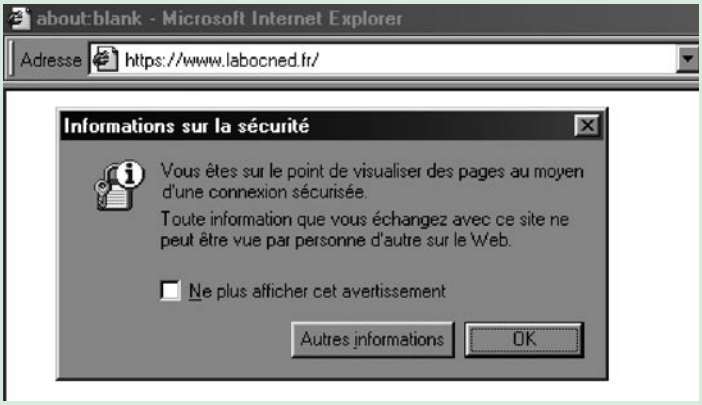

about:blank - Microsoft Internet Explorer
Adresse
https://www.labocned.fr/
Informations sur la sécurité
Vous êtes sur le point de visualiser des pages au moyen d'une connexion sécurisée.
Toute information que vous échangez avec ce site ne peut être vue par personne d'autre sur le Web.
Ne plus afficher cet avertissement
Autres informations
OK

Figure 5 : avertissement https

Un avertissement nous accueille. Lorsque l'on fait OK, on arrive sur la page :

Figure 6 : page d'accueil

Si on clique dessus, on peut consulter les informations du certificat. En parcourant les différentes rubriques, vous verrez qu'il contient les informations suivantes :

Un Certificat Serveur contient les informations suivantes :
• Le nom de domaine du serveur : www.societe.fr
• Quelques informations optionnelles (nom et adresse physique de la société, etc.)
• La clé publique du serveur à certifier
• La date d'expiration du certificat
• Un numéro de série unique
• Le nom de l'Autorité de Certification qui a délivré le certificat numérique (Verisign Trust Network)
• La signature de l'Autorité de Certification qui a délivré le certificat numérique (au moyen de la clé secrète de l'Autorité de Certification, sorte de "tampon" de validation d'un passeport électronique).

Figure 7 : rubrifiques d'un certificat

Le telnet chiffré : SSH

Vous allez maintenant faire la manipulation tout seul ou presque…
http://formation-debian.via.ecp.fr/ch35.html

Vous connaissez telnet, nous l'avons utilisé à maintes reprises. Je vous avais déconseillé, à l'époque, de l'utiliser en entreprise car celui-ci n'est absolument pas sécurisé. Un petit malin sur votre réseau, avec un logiciel de capture de trames peut très facilement récupérer les informations que vous tapez, en particulier votre mot de passe.
Le telnet sécurisé que vous allez mettre en œuvre, s'appelle SSH. Avec lui, vous n'avez rien à craindre❶.

Exercice 6

Vous êtes dans votre dernier tome de cours, vous devez apprendre à travailler en autonomie. Je ne vais donc pas vous donner directement les commandes. Vous allez vous débrouiller tout seul. Voici seulement les consignes.
Votre mission consiste à mettre en œuvre un serveur SSH sur votre serveur Linux (le serveur s'appelle sshd et vous aurez probablement des paquetages à installer). Ensuite, vous installerez un client sur votre machine Windows.
À l'heure actuelle, Windows n'intègre pas de client compatible SSH. Je vous conseille d'utiliser le logiciel libre putty (http://www.chiark.greenend.org.uk/~sgtatham/putty/).
Enfin, vous répondrez à la question suivante : sur quel port écoute le serveur sshd ?

❶ *Enfin… on n'est jamais trop prudent non plus. Un administrateur de réseau qui se respecte à toujours tendance à être un paranoïaque.*

Le chiffrement des données permet de les rendre illisibles à qui n'est pas autorisé. La méthode à « clés asymétriques » est la plus répandue. La clé publique du destinataire sert à chiffrer. La clé privée du destinataire lui sert à déchiffrer.

L'algorithme le plus répandu est SSL. Il sert à sécuriser les protocoles applicatifs tels http, ldap, etc.

Si vous voulez approfondir

Vous pouvez vous documenter sur le fonctionnement des protocoles de chiffrement comme RSA, Idea, Blowfish, etc.

Vous pouvez vous intéresser au chiffrement d'autres protocoles applicatifs (ftps, pops, imaps, etc.)

Consultez http://www.gnupgp.org pour voir comment on peut chiffrer et signer des messages électroniques.

Mise en place d'un annuaire LDAP

Durée indicative : 3 heures

▶ *Objectif*

Mettre en œuvre le gestionnaire d'annuaire OpenLDAP afin de partager une base d'informations sur les utilisateurs, les matériels et les logiciels du système informatique.

▶ *Conditions préalables*

Aucune en particulier

▶ *Considérations techniques*

Aucune en particulier

▶ *Liens*

http://www.openldap.org/
http://www.coagul.org/article.php3?id_article=172
http://www-id.imag.fr/~svarrett/download/polys/Tutorial_LDAP_HTML/index.html

Introduction

Imaginez que vous êtes le patron d'une petite entreprise. Vous décidez de vous informatiser. Vous achetez quelques ordinateurs isolés que vous mettez sur le bureau de quelques employés. Ceux-ci utilisent des produits bureautiques, des logiciels de gestion standards et une application « métier » très spécifique. Il leur faut saisir un mot de passe pour ouvrir une session sur leur machine puis un autre mot de passe pour leur logiciel de comptabilité et encore un autre mot de passe pour leur application. Ensuite, vous décidez de connecter les ordinateurs en réseau. Vous partagez des imprimantes, des unités de sauvegarde, vos utilisateurs créent des dossiers partagés pour s'échanger des fichiers.
Pendant ce temps-là, vos affaires prennent de l'ampleur, le nombre de vos fournisseurs augmente ainsi que celui de vos clients. Lorsque vous cherchez à contacter un fournisseur, il faut chercher leurs coordonnées dans un répertoire, ou dans un agenda ou regarder sur une facture ou encore consulter leur fichier dans votre logiciel de gestion. Idem pour vos clients.
Puis vous décidez de connecter votre entreprise à Internet, vous pouvez ainsi expédier et recevoir des messages. Mais où sont stockées les adresses emails de vos correspondants : sur une machine, sur plusieurs ? Sur le poste de la secrétaire ou celui du comptable ?

Continuons. Vos affaires marchent très bien. Vous rachetez un de vos concurrents. Pour faire des économies, vous décidez de fusionner certains services. En particulier, vous nommez un informaticien chef chargé de piloter l'informatique dans les deux entreprises qui n'en font plus qu'une seule. Un utilisateur de l'autre entreprise vous appelle car il a un problème d'impression sur son poste, mais quelle est son adresse IP, sa configuration, l'imprimante qu'il utilise habituellement, etc. ?

Passons. Ensuite vous décidez de mettre en place un Extranet avec vos fournisseurs, de nouveaux problèmes apparaissent : à quelles applications ont-ils accès, avec quels droits, quels mots de passe, etc.

Je pourrais continuer longtemps cette histoire. L'idée est de vous sensibiliser au fait qu'il devient nécessaire dans certaines entreprises de mettre en place un système de gestion centralisé des utilisateurs, des contacts, des ressources, etc. Ce système s'appelle un **annuaire.**

Dans la vraie vie, qu'est-ce qu'un annuaire ? Tout simplement, un endroit central où l'on peut trouver des informations sur des gens. Par exemple :

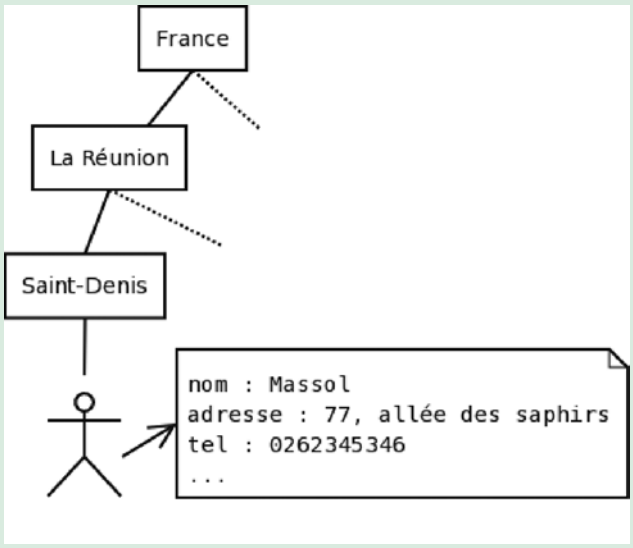


Figure 1 : un exemple d'annuaire classique

On voit apparaître une structure hiérarchique. Dans l'exemple donné, lorsque je prends l'annuaire de mon département, il est sous-entendu que cela fait partie des annuaires français. Ensuite, les informations sont classées par ville puis par nom. Lorsque l'on a trouvé le nom, on obtient l'adresse et le numéro de téléphone.

Voici quelques observations :

– certains éléments (la ville par exemple) servent à organiser les informations ;
– certains éléments sont porteurs d'information (par exemple, le numéro de téléphone associé à chaque personne dans l'annuaire) ;
– pour être véritablement identifié, il faut dire « Massol du 77, allée des saphirs à Saint-Denis de La Réunion en France ».

Les annuaires LDAP

Passons maintenant à un annuaire informatique. LDAP (lightweight directory access protocol) est un protocole internet client-serveur largement répandu. En effet, les principaux constructeurs informatiques l'ont mis en œuvre ou s'en sont inspirés (que ce soit Novell avec son NDS ou Microsoft avec Active Directory[1]).

• Présentation

Un annuaire LDAP sert à stocker des informations sur des gens (utilisateurs ou contacts) mais aussi sur des appareils ou des logiciels. Par exemple :

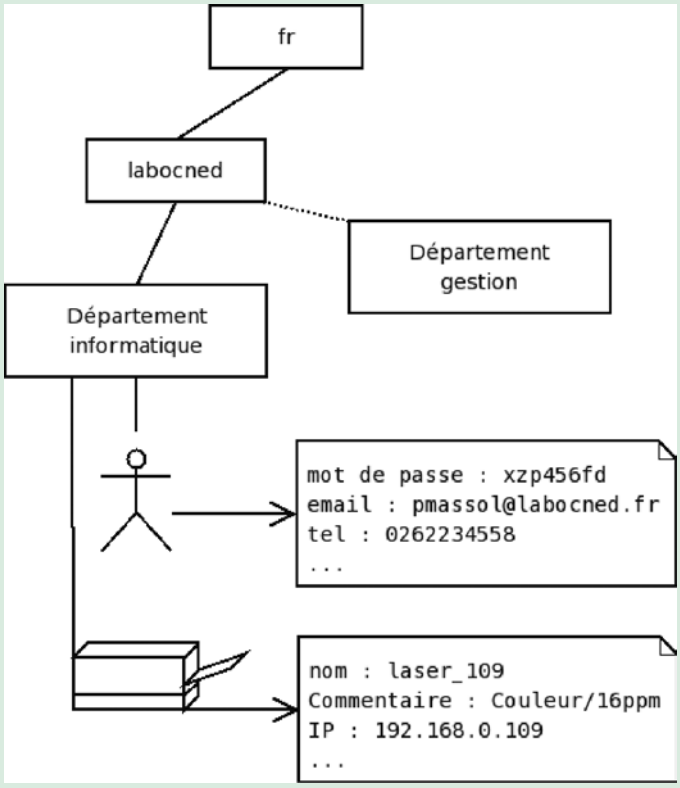


Figure 2 : un exemple d'annuaire LDAP

La racine (**root**) de cet annuaire correspond à notre nom de domaine : labocned.fr. Les départements servent à organiser les objets contenus dans notre annuaire, en langage LDAP, on appelle ça des **unités d'organisation** (souvent appelées OU pour *organisational unit*). Les objets contenus dans les unités d'organisation sont décrits par un certain nombre de données que l'on appelle **attributs**.

[1] *Comme à son habitude, cet éditeur s'est très largement inspiré du protocole mais, comme par hasard, son produit n'est pas conforme à la RFC définissant LDAP.*

• Installation des paquetages nécessaires

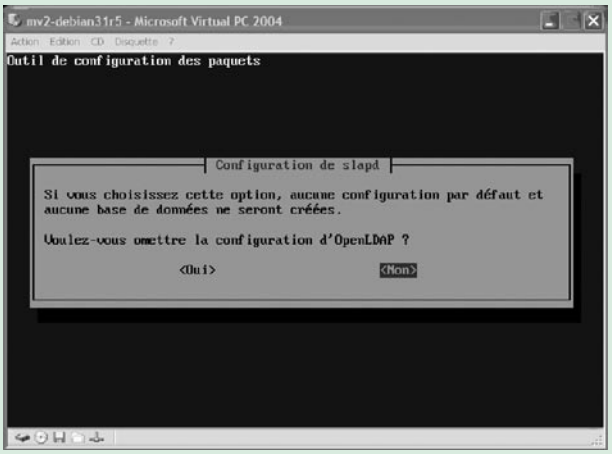
Nous allons travailler avec une implémentation libre de LDAP : *openldap*.

http://www.openldap.org

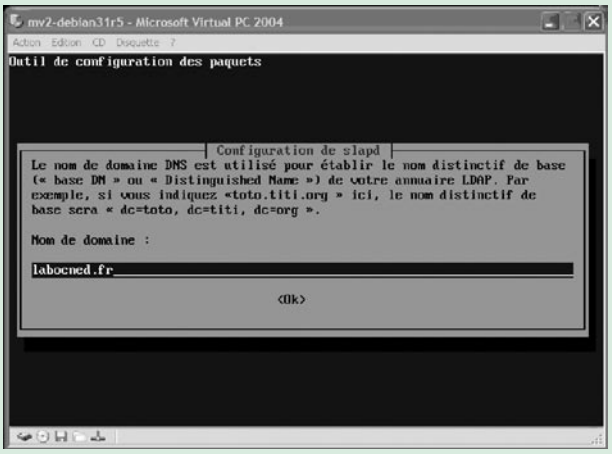
Installation des paquetages :

```
mv2-debian:/# apt-get install ldap-server ldap-client
```

Ce qui aura pour effet d'installer les paquets suivants :
slapd ldap-utils et des librairies…
Suite au dépaquetage quelques écrans de première configuration vous seront proposés :

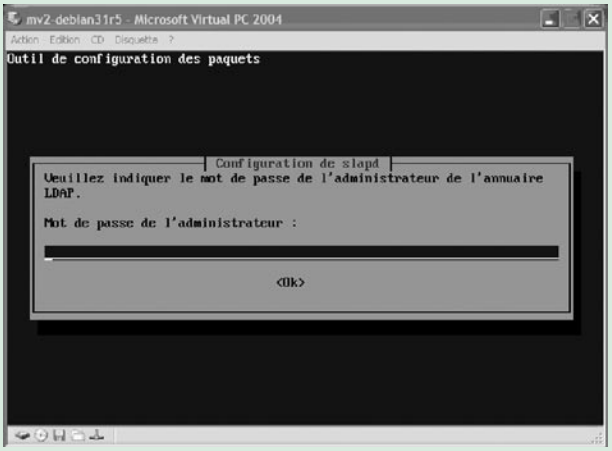

mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de slapd
Si vous choisissez cette option, aucune configuration par défaut et aucune base de données ne seront créées.
Voulez-vous omettre la configuration d'OpenLDAP ?
<Oui>
<Non>

Configuration de slapd
Attention à la formulation de la question :
« Voulez-vous omettre… »
Nous, nous voulons :
Choix = <Non> car nous allons tout configurer à la main pour mieux comprendre l'organisation et le fonctionnement de LADP ; mais ce qui peut être fait maintenant…

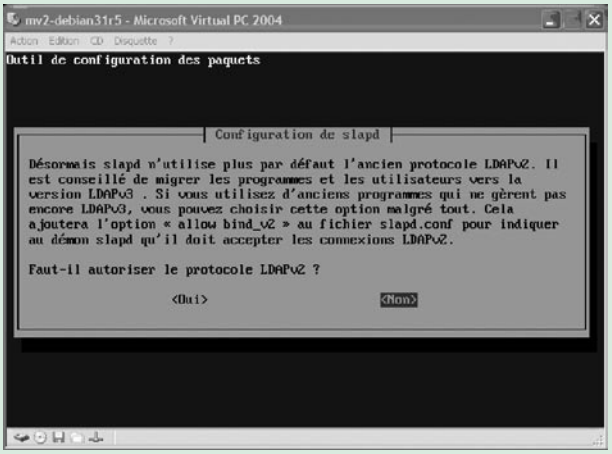

mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de slapd
Le nom de domaine DNS est utilisé pour établir le nom distinctif de base (« base DN » ou « Distinguished Name ») de votre annuaire LDAP. Par exemple, si vous indiquez «toto.titi.org » ici, le nom distinctif de base sera « dc=toto, dc=titi, dc=org ».
Nom de domaine :
labocned.fr
<Ok>

Nom de domaine = **labocned.fr**

Nom de l'entité d'organisation = **labo-cned.fr**


mv2-debian31r5 - Microsoft Virtual PC 2004
Action Edition CD Disquette ?
Outil de configuration des paquets
Configuration de slapd
Veuillez indiquer le mot de passe de l'administrateur de l'annuaire LDAP.
Mot de passe de l'administrateur :
<Ok>

Mot de passe de l'administrateur de l'annuaire LDAP
Mettez un mot de passe spécifique LDAP bien sûr, comme :
admin-ldap-labocned


mv2-debian31r5 - Microsoft Virtual PC 2004
Action Edition CD Disquette ?
Outil de configuration des paquets
Configuration de slapd
Désormais slapd n'utilise plus par défaut l'ancien protocole LDAPv2. Il est conseillé de migrer les programmes et les utilisateurs vers la version LDAPv3 . Si vous utilisez d'anciens programmes qui ne gèrent pas encore LDAPv3, vous pouvez choisir cette option malgré tout. Cela ajoutera l'option « allow bind_v2 » au fichier slapd.conf pour indiquer au démon slapd qu'il doit accepter les connexions LDAPv2.
Faut-il autoriser le protocole LDAPv2 ?
<Oui>
<Non>

Configuration de sldapd
Autorisez le protocole ldap v2, ceci est valable en particulier pour les OS de type windows 2000.

• Configuration

Le fichier *slapd.conf*, qui se trouve dans */etc/ldap*, contient les informations de configuration nécessaires à votre serveur LDAP *slapd*. Il vous faudra éditer ce fichier pour le rendre spécifique à votre domaine.
Voici à quoi il ressemble avant de le configurer :

```
# Allow LDAPv2 binds
allow bind_v2
# This is the main slapd configuration file [...]
# Global Directives
[...]
# Schema and objectClass definitions
include     /etc/ldap/schema/core.schema
include     /etc/ldap/schema/cosine.schema
include     /etc/ldap/schema/nis.schema
include     /etc/ldap/schema/inetorgperson.schema

# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile    /var/run/slapd/slapd.args

# Read slapd.conf(5) for possible values
logleval     0

# Where the dynamically loaded modules are stored
modulepath /usr/lib/ldap
moduleload  back_bdb

# The maximum number of entries that is returned for a search operation
sizelimit 500

# The tool-threads parameter sets the actual amount of cpu's that is used
# for indexing.
Tool-threads 1

########################################################################
# Specific Directives for database #1, of type bdb:
# Database specific directives apply to this database until another
# 'database' directives occurs
database  bdb
# The base of your directory in database #1
suffix     "dc=labocned,dc=fr"
```

```
# rootdn directive for specifying a superuser on the database. This is needed for syncrepl.
# rootdn        "cn=admin,dc=labocned,dc=fr"
# Where the database file are physically stored for databse #1
directory       "/var/lib/ldap"
```

La ligne de suffixe nomme le domaine pour lequel le serveur LDAP fournira les informations. Nous aurons donc :

```
suffix        "dc=labocned, dc=fr"
```

dc signifie *domain component*. L'ensemble des dc correspond généralement au nom de domaine DNS.
L'entrée *rootdn* décrit l'administrateur de l'annuaire. Décommentez cette ligne ainsi :

```
rootdn        «cn=admin, dc=labocned, dc=fr»
```

cn signifie *common name* et correspond au nom de l'objet.
Il faut ensuite générer un mot de passe pour cet admin. Ajoutez la ligne :

```
rootpw      <votre mot de passe>
```

Attention ! Là il sera en clair. Il est donc préférable d'en générer un crypté et ensuite de le recopier. Prenons par exemple le mot de passe **labocned**

```
mv2-debian/etc/ldap# slappasswd
New password : <tapez labocned>
Re-enter new password: <re-tapez labocned>
{SSHA}ACSuFlmjG1Yy1zqx8w/f7Qs7KDVoz2ds
```

C'est donc cette chaîne tout à fait compréhensible que vous indiquerez à **rootpw.**

Toujours dans **slapd.conf** vérifiez les droits sur la base :

```
access to attrs=userPassword,shadowLastChange
      by dn="cn=admin,dc=labocned,dc=fr" write
      by anonymous auth
      by self write
      by * none
[...]
access to dn.base="" by * read
[...]
access to *
      by dn="cn=admin,dc=labocned,dc=fr" write
      by * read

[...ajouter...]
# Tous les utilisateurs authentifiés peuvent modifier certaines valeurs d'attributs
access to attrs=userPassword,sambaLMPassword,sambaNTPassword,description
      by self write
      by anonymous auth
      by * none
```

Vérifiez que les droits affectés au répertoire de travail (directive *directory* dans le fichier *slapd.conf*) de LDAP sont corrects :

```
mv2-debian:/# ls /var/lib –la
[...]
drwxr-xr-x2 openldap openldap 4096 2007-07-20 12:50 ldap
[...]
```

Relancez le service :

```
mv2-debian/etc/ldap:# /etc/init.d/slapd restart
```

Le démon LDAP étant exécuté par le pseudo-utilisateur *openldap* :

```
mv2-debian:/etc/ldap# ps aux | grep slapd
openldap    2818 0.0 1.4 14424 3796 ?    Ssl    14:05 0:00 /usr/sbin/slapd -g openldap –u openldap
```

Celui-ci doit avoir tous les droits sur le répertoire. Il est bon également que root puisse écrire dans ce répertoire si vous faites des importations en bloc avec la commande *slapadd*, par exemple.

(Re)démarrez le démon slapd :

```
mv2-debian:/etc/ldap# /etc/init.d/sldapd restart
```

Enfin, lançons une incantation qui nous permettra de vérifier le bon fonctionnement :

```
mv2-debian:/etc/ldap# ldapsearch -x -b '' -s base '(objectclass=*)'
# extended LDIF
#
# LDAPv3
# base <> with scope baseObject
# filter: (objectclass=*)
# requesting: ALL
#


#
dn:
objectClass: top
objectClass: OpenLDAProotDSE

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

• Les principales commandes

Le paquetage OpenLDAP contient deux démons : *slapd* et *slurpd* :

- le démon *slapd* est le démon LDAP autonome que vous devez exécuter pour prendre en charge LDAP ;
- le démon *slurpd* contrôle la duplication des annuaires LDAP sur un réseau. Nous ne mettrons pas en œuvre cette fonctionnalité.

OpenLDAP contient également certains utilitaires pour l'ajout, la modification et la suppression d'entrées dans un annuaire LDAP :

- *ldapmodify* : modifie des entrées dans l'annuaire ;
- *ldapadd* : ajoute des entrées à l'annuaire ;
- *ldapsearch* : recherche des entrées dans l'annuaire ;
- *ldapdelete* : efface des entrées dans l'annuaire.

• Création de la racine

Créons la racine de notre annuaire ainsi que l'enregistrement correspondant à l'administrateur. Avec votre éditeur favori, tapez le fichier **societe.txt** ci-dessous :

```
mv2-debian:/etc/ldap# vi societe.txt
# La racine de l'annuaire
dn: dc=labocned,dc=fr
dc: labocned
objectClass: top
objectClass: domain
objectClass: domainRelatedObject
description: Notre labo du CNED !
associatedDomain: labocned.fr

dn: cn=root,dc=labocned,dc=fr
objectClass: organizationalRole
cn: root
```

Donnons quelques explications :

- dn (*distinguished name*) est un identifiant. Il correspond à la chaîne complète permettant de retrouver une information dans l'annuaire (voir plus loin). On note ici que le dn correspond à la racine ;
- *objectclass* définit une sorte de « dictionnaire des données[1] » pour l'objet que nous sommes en train de créer. Il indique quels attributs peuvent être associés à l'objet. Il existe de nombreuses classes d'objet que l'on utilise en fonction des besoins. Par exemple, lorsque l'on crée un utilisateur, on utilisera un *objectclass* différent de celui pour une imprimante.

Puis à l'aide de la commande *ldapadd*, ajoutons ces informations à l'annuaire :

```
mv2-debian:/etc/ldap# ldapadd -x -D "cn=admin,dc=labocned,dc=fr" -W -f societe.txt
Enter LDAP Password: <tapez votre mot de passe en clair>
adding new entry "dc=labocned,dc=fr"
```

La commande *ldapadd* contient un certain nombre de paramètres :

- x : afin de ne pas utiliser l'authentification sasl (par défaut, semble-t-il) ;
- D : indique que nous allons faire figurer le dn de l'administrateur (cn=root, dc=labocned,dc=fr) ;
- W : afin que le logiciel nous demande le mot de passe de l'administrateur (vous saisirez le mot de passe (en clair) que vous avez créé lors de la configuration du fichier sladp.conf) ;
- f : pour indiquer le fichier qui contient les informations à ajouter.

[1] *Si ça vous rappelle votre cours d'analyse MERISE, vous êtes tout à fait dans le coup !*

• Création d'une unité d'organisation

Afin de bien organiser nos données, nous allons maintenant créer une unité d'organisation que nous appellerons « pédagogie ».
Les OU reflètent généralement la structure de l'entreprise (les services). Ensuite, dans une OU on peut créer d'autres OU ou des objets en rapport avec le service (ses groupes d'utilisateurs, ses utilisateurs, ses contacts, ses matériels, etc.).
Toujours avec votre éditeur de texte favori, créez un fichier ou1.txt :

```
mv2-debian:/etc/ldap# vi ou1.txt
dn: ou=pedagogie, dc=labocned, dc=fr
objectclass: organizationalUnit # objet définissant un OU
ou: pedagogie       # nom de l'OU
```

Ensuite, nous l'ajoutons à la base :

```
mv2-debian:/etc/ldap# ldapadd -x -D «cn=admin,dc=labocned,dc=fr» -W -f ou1.txt
Enter LDAP Password:
adding new entry "ou=pedagogie, dc=labocned, dc=fr"
```

Exercice 7

Nous avons déjà utilisé la commande ldapsearch. À partir de l'aide en ligne ou de toute autre source, trouvez la commande à taper afin d'obtenir le nom de l'OU créée (et rien que ça !).

• Création d'un groupe d'utilisateurs

Nous allons créer un groupe d'utilisateurs que nous placerons dans l'OU pédagogie.
Toujours avec votre éditeur de texte favori, créez un fichier groupe1.txt :

```
mv2-debian:/etc/ldap# vi group1.txt
dn: cn=utilisateurs,ou=pedagogie,dc=labocned,dc=fr
cn: utilisateurs        # nom du groupe
objectClass: posixgroup  # objet définissant un groupe
gidNumber: 600          # un numéro arbitraire
```

Ensuite, nous l'ajoutons à la base :

```
mv2-debian:/etc/ldap# ldapadd -x -D «cn=admin, dc=labocned, dc=fr» -W -f group1.txt
Enter LDAP Password:
adding new entry "cn=utilisateurs,ou=pedagogie,dc=labocned,dc=fr"
```

• Création d'un utilisateur

Nous allons ajouter un utilisateur au groupe d'utilisateurs créé précédemment et nous le placerons également dans l'OU correspondante. Voici les caractéristiques de l'utilisateur Paul HOCHON :

```
mv2-debian:/etc/ldap# vi util1.txt
dn: cn=phochon,ou=pedagogie,dc=labocned,dc=fr
uid: phochon                    # identifiant de login
cn: phochon                     # common name
sn: Hochon                      # nom de famille
objectclass: top                # liste des objets utilisés
objectclass: person
objectclass: posixAccount
objectclass: inetOrgPerson
uidNumber: 1234                 # un numéro arbitraire
gidNumber: 600                  # numéro du groupe précédent
homeDirectory: /home/phochon
loginShell: /bin/bash
mail: phochon@labocned.fr
givenName: Paul
```

Vous constatez que de nombreux attributs sont associés à l'objet. On peut cumuler les classes d'objet afin de l'utiliser de différentes façon. Par exemple, ici, nous prenons la classe d'objet *posixAccount* afin de permettre à notre utilisateur de se connecter à la machine (d'où tous les attributs qui vous rappellent le fichier /etc/passwd) et nous prenons aussi *inetOrgPerson* car je veux pouvoir consulter l'annuaire à partir de mon logiciel de messagerie (attribut *mail* en particulier).
Comment connaître les classes d'objet ? Dans cet atelier, il n'est pas question de détailler tous les possibilités de LDAP et tous les objets. Voyez la section « Pour approfondir ».
Ensuite, nous l'ajoutons à la base :

```
mv2-debian:/etc/ldap# ldapadd -x -D «cn=admin, dc=labocned, dc=fr» -W -f util1.txt
Enter LDAP Password:
adding new entry "cn=phochon,ou=pedagogie,dc=labocned,dc=fr"
```

• Ajout d'un mot de passe

Si je veux que mon utilisateur puisse se connecter, il lui faut un mot de passe :

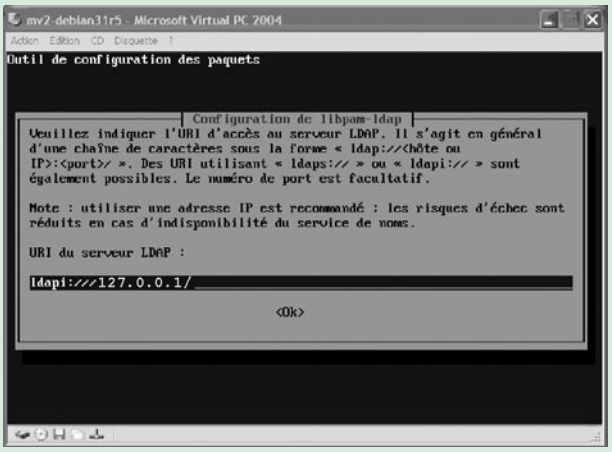
```
mv2-debian:/etc/ldap# ldappasswd -x -D «cn=admin, dc=labocned, dc=fr» -W «cn=phochon,ou=pe
dagogie, dc=labocned, dc=fr» -s mon_mot_de_passe
Enter bind password:
Result: Success (0)
```

Il faut indiquer correctement le dn : "cn=phochon,ou=pedagogie, dc=labocned, dc=fr" de notre utilisateur.

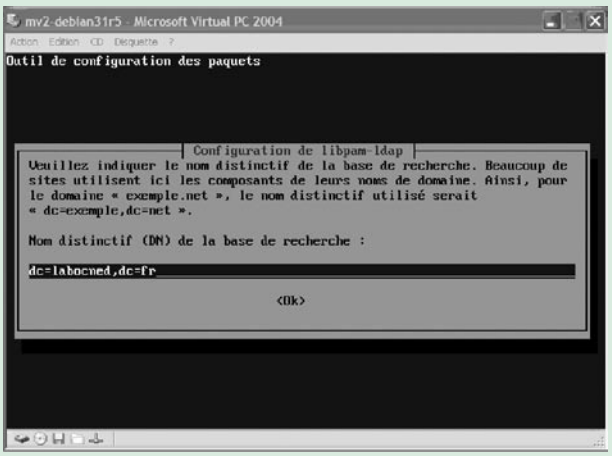
Connexion de l'utilisateur

Nous voulons que notre utilisateur Paul HOCHON puisse se connecter comme un utilisateur « classique » (souvenez-vous des premiers TP !). Mais là, ça se complique un petit peu. Jusqu'à présent, lorsqu'un utilisateur se connecte, le processus *login* demande à l'utilisateur de saisir un nom et un mot de passe puis vérifie sa validité avec le fichier /etc/passwd. Ici, Paul HOCHON n'existe pas dans /etc/passwd (vérifiez-le si vous ne me croyez pas !). Il a été créé dans l'annuaire. Il faut donc expliquer aux mécanismes de *login* d'aller en premier dans /etc/passwd, et au cas où l'utilisateur n'y figure pas, d'aller voir dans l'annuaire.
http://wiki.debian.net/?LDAPAuthentication
PAM (Pluggable Authentication Modules) permet un mode d'authentification modulaire facilement paramétrable. Il existe donc un module PAM pour LDAP, soit :

```
mv2-debian:/etc# apt-get install libpam-ldap
```


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libpam-ldap
Veuillez indiquer l'URI d'accès au serveur LDAP. Il s'agit en général d'une chaîne de caractères sous la forme « ldap://<hôte ou IP>:<port>/ ». Des URI utilisant « ldaps:// » ou « ldapi:// » sont également possibles. Le numéro de port est facultatif.
Note : utiliser une adresse IP est recommandé : les risques d'échec sont réduits en cas d'indisponibilité du service de noms.
URI du serveur LDAP :
ldapi:///127.0.0.1/
<Ok>

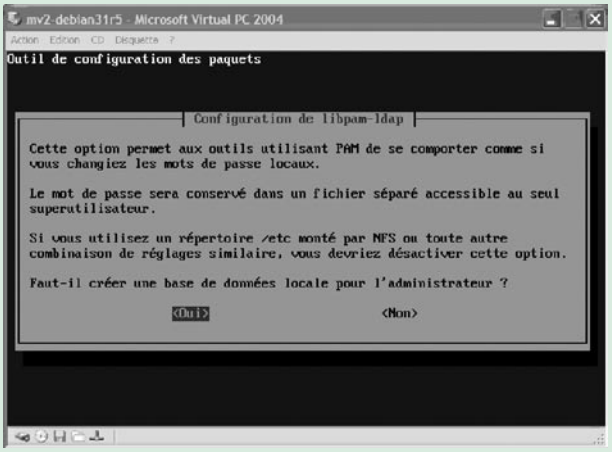
Renseigner l'adresse IP du serveur LDAP://127.0.0.1/


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libpam-ldap
Veuillez indiquer le nom distinctif de la base de recherche. Beaucoup de sites utilisent ici les composants de leurs noms de domaine. Ainsi, pour le domaine « exemple.net », le nom distinctif utilisé serait « dc=exemple,dc=net ».
Nom distinctif (DN) de la base de recherche :
dc=labocned,dc=fr
<Ok>

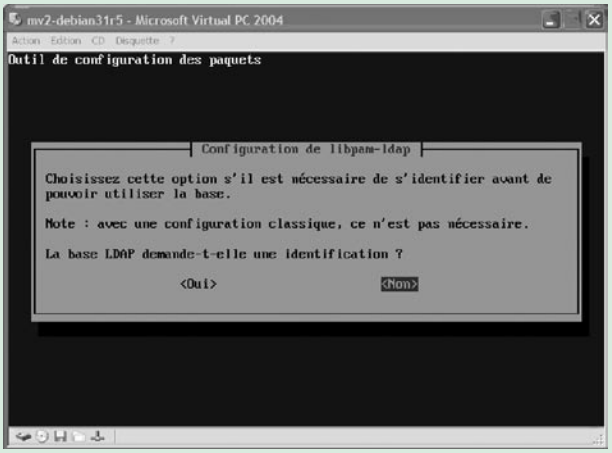
Renseigner le dn de la base : dc=labocned,dc=fr

mv2-debian31r5 - Microsoft Virtual PC 2004
Action Edition CD Disquette ?
Outil de configuration des paquets
Configuration de libpam-ldap
Veuillez indiquer la version du protocole LDAP que doit utiliser ldapns. Il est recommandé de choisir le numéro de version le plus élevé disponible.
Version de LDAP à utiliser :
3
2
<Ok>

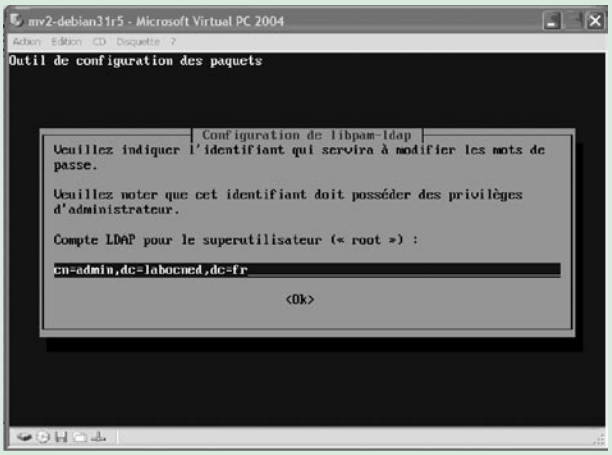
Nous sommes en LDAPv3


mv2-debian31r5 - Microsoft Virtual PC 2004
Action Edition CD Disquette ?
Outil de configuration des paquets
Configuration de libpam-ldap
Cette option permet aux outils utilisant PAM de se comporter comme si vous changiez les mots de passe locaux.
Le mot de passe sera conservé dans un fichier séparé accessible au seul superutilisateur.
Si vous utilisez un répertoire /etc monté par NFS ou toute autre combinaison de réglages similaire, vous devriez désactiver cette option.
Faut-il créer une base de données locale pour l'administrateur ?
<Oui>
<Non>

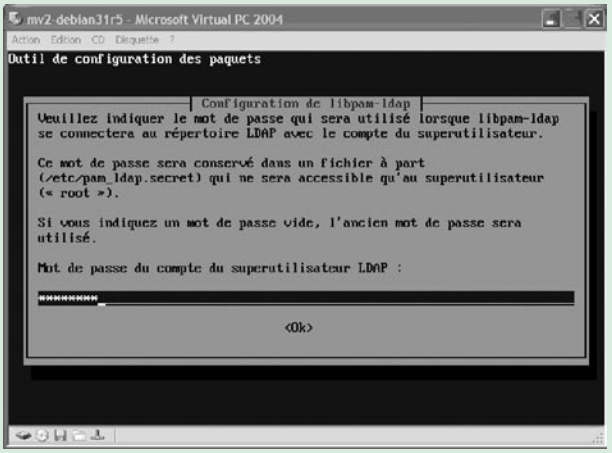
<Oui> créer une base locale


mv2-debian31r5 - Microsoft Virtual PC 2004
Action Edition CD Disquette ?
Outil de configuration des paquets
Configuration de libpam-ldap
Choisissez cette option s'il est nécessaire de s'identifier avant de pouvoir utiliser la base.
Note : avec une configuration classique, ce n'est pas nécessaire.
La base LDAP demande-t-elle une identification ?
<Oui>
<Non>

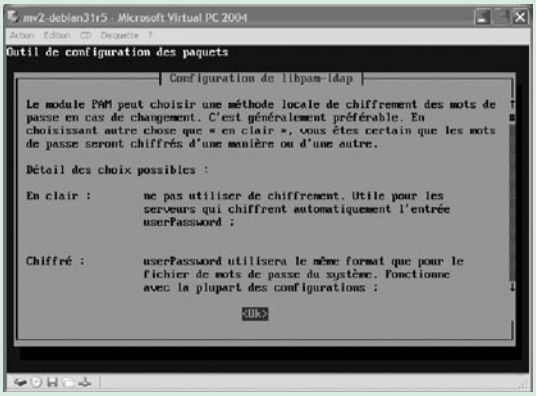
<Oui> la base LDAP demande une authentification (ils ne parlent pas de l'administrateur de la base…)

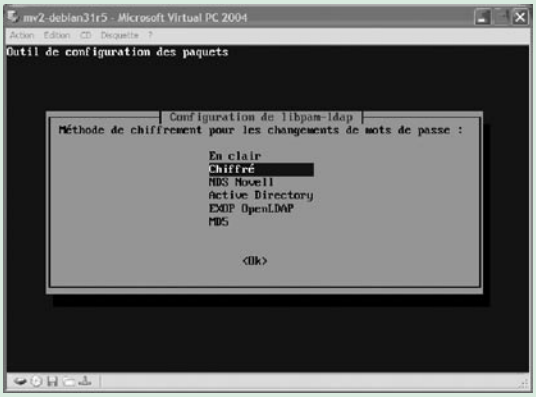

mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libpam-ldap
Veuillez indiquer l'identifiant qui servira à modifier les mots de passe.
Veuillez noter que cet identifiant doit posséder des privilèges d'administrateur.
Compte LDAP pour le superutilisateur (« root ») :
cn=admin,dc=labocned,dc=fr
<Ok>

Ici il faut renseigner le compte de l'administrateur LDAP : cn=admin,dc=labocned,dc=fr


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libpam-ldap
Veuillez indiquer le mot de passe qui sera utilisé lorsque libpam-ldap se connectera au répertoire LDAP avec le compte du superutilisateur.
Ce mot de passe sera conservé dans un fichier à part (/etc/pam_ldap.secret) qui ne sera accessible qu'au superutilisateur (« root »).
Si vous indiquez un mot de passe vide, l'ancien mot de passe sera utilisé.
Mot de passe du compte du superutilisateur LDAP :
****
<Ok>

… son mot de passe.


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libpam-ldap
Le module PAM peut choisir une méthode locale de chiffrement des mots de passe en cas de changement. C'est généralement préférable. En choisissant autre chose que « en clair », vous êtes certain que les mots de passe seront chiffrés d'une manière ou d'une autre.
Détail des choix possibles :
En clair : ne pas utiliser de chiffrement. Utile pour les serveurs qui chiffrent automatiquement l'entrée userPassword ;
Chiffré : userPassword utilisera le même format que pour le fichier de mots de passe du système. Fonctionne avec la plupart des configurations ;
<Ok>



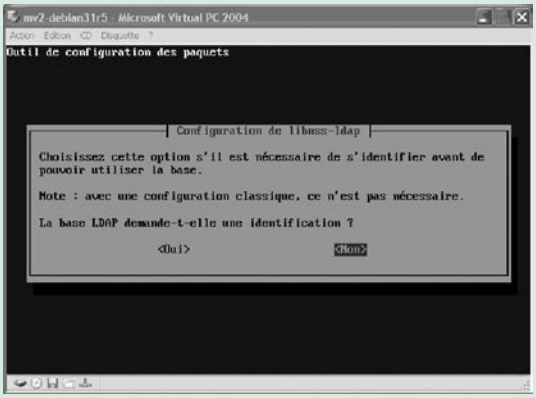
Choisissez <Chiffré>

NSS: Name Service Switch
NSS est le service qui permet de faire abstraction de la façon dont sont stockées les informations concernant les utilisateurs, les groupes etc. C'est ce que nous voulons : que les utilisateurs aient été créés de façon « standard » (useradd…) ou via ldap (fichier ldif…). Nous voulons qu'ils puissent s'authentifier de la même façon.

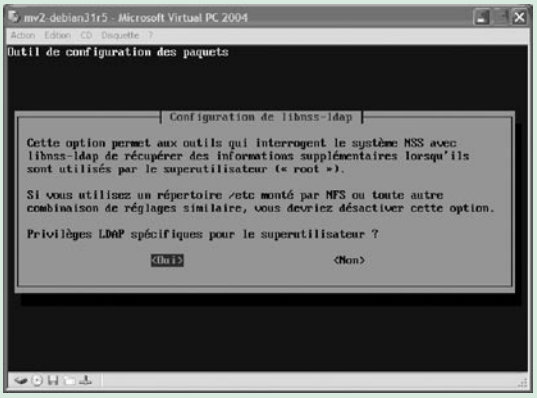
libnss – librairie name service switch

Un programme (login, passwd,...) ayant besoin de ce genre d'infos doit savoir comment accéder aux fichiers : où ils sont stockés, comment ils sont organisés... La librairie NSS permet de s'affranchir de cette contrainte. Le programme ne sait pas où sont stockées les informations, il a juste un ensemble de fonctions à sa disposition pour manipuler les données. Cela permet, entre autre, de délocaliser les comptes de /etc/passwd vers une base de données ou... un annuaire ldap bien sûr sans avoir à retoucher les programmes ! La librairie se charge d'aller chercher les infos au bon endroit ; elle gère de nombreuses « bases de données » : les comptes utilisateurs, les groupes, les machines, les services, les services rpc, les clé publiques... (cf *man nsswitch.conf*)
Nous allons maintenant installer et configurer le **name service** pour que le système puisse aller interroger ldap si nécessaire :

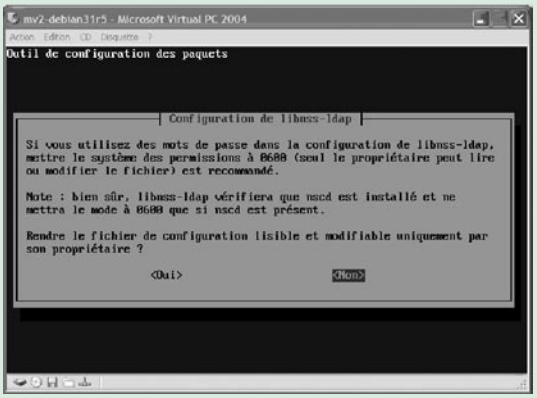
```
mv2-debian:/etc# apt-get install libnss-ldap
```


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libnss-ldap
Choisissez cette option s'il est nécessaire de s'identifier avant de pouvoir utiliser la base.
Note : avec une configuration classique, ce n'est pas nécessaire.
La base LDAP demande-t-elle une identification ?
<Oui>
<Non>

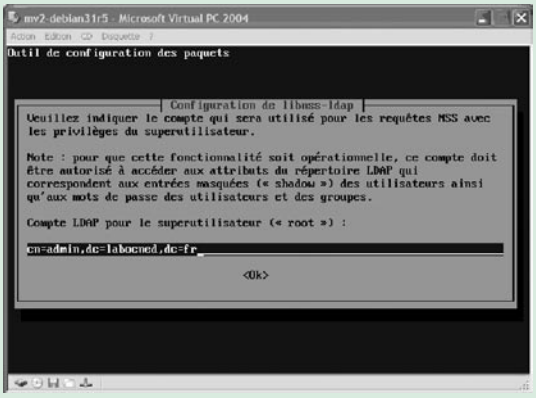
<Oui> la base LDAP demande une identification


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libnss-ldap
Cette option permet aux outils qui interrogent le système NSS avec libnss-ldap de récupérer des informations supplémentaires lorsqu'ils sont utilisés par le superutilisateur (« root »).
Si vous utilisez un répertoire /etc monté par NFS ou toute autre combinaison de réglages similaire, vous devriez désactiver cette option.
Privilèges LDAP spécifiques pour le superutilisateur ?
<Oui>
<Non>

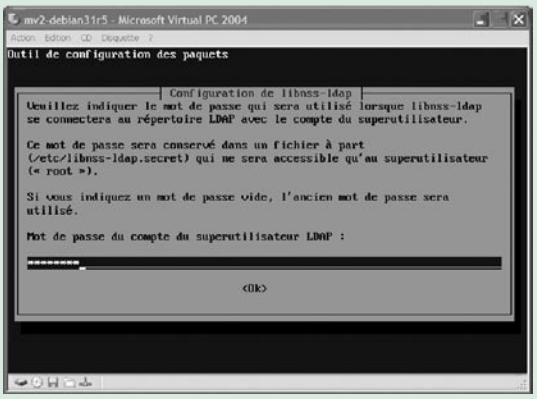
<Oui> le superutilisateur aura des privilèges spécifiques


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libnss-ldap
Si vous utilisez des mots de passe dans la configuration de libnss-ldap, mettre le système des permissions à 0600 (seul le propriétaire peut lire ou modifier le fichier) est recommandé.
Note : bien sûr, libnss-ldap vérifiera que nscd est installé et ne mettra le mode à 0600 que si nscd est présent.
Rendre le fichier de configuration lisible et modifiable uniquement par son propriétaire ?
<Oui>
<Non>

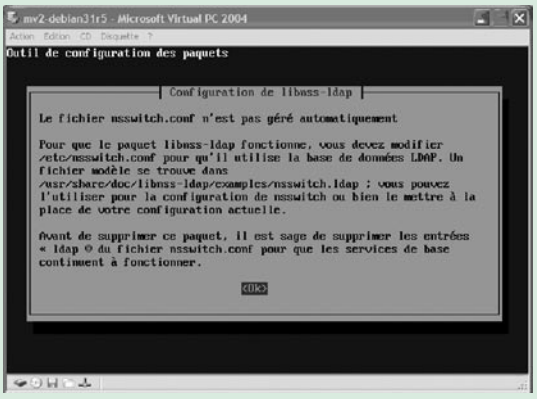
<Non> ne soyons pas paranoïaque…


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libnss-ldap
Veuillez indiquer le compte qui sera utilisé pour les requêtes NSS avec les privilèges du superutilisateur.
Note : pour que cette fonctionnalité soit opérationnelle, ce compte doit être autorisé à accéder aux attributs du répertoire LDAP qui correspondent aux entrées masquées (« shadow ») des utilisateurs ainsi qu'aux mots de passe des utilisateurs et des groupes.
Compte LDAP pour le superutilisateur (« root ») :
cn=admin,dc=labocned,dc=fr
<Ok>

Là vous savez quoi répondre…


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libnss-ldap
Veuillez indiquer le mot de passe qui sera utilisé lorsque libnss-ldap se connectera au répertoire LDAP avec le compte du superutilisateur.
Ce mot de passe sera conservé dans un fichier à part (/etc/libnss-ldap.secret) qui ne sera accessible qu'au superutilisateur (« root »).
Si vous indiquez un mot de passe vide, l'ancien mot de passe sera utilisé.
Mot de passe du compte du superutilisateur LDAP :
<Ok>

Ici aussi…


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de libnss-ldap
Le fichier nsswitch.conf n'est pas géré automatiquement
Pour que le paquet libnss-ldap fonctionne, vous devez modifier /etc/nsswitch.conf pour qu'il utilise la base de données LDAP. Un fichier modèle se trouve dans /usr/share/doc/libnss-ldap/examples/nsswitch.ldap ; vous pouvez l'utiliser pour la configuration de nsswitch ou bien le mettre à la place de votre configuration actuelle.
Avant de supprimer ce paquet, il est sage de supprimer les entrées « ldap » du fichier nsswitch.conf pour que les services de base continuent à fonctionner.
<Ok>

Enfin notez bien qu'il nous faudra réaliser une petite modification manuelle…

Nous avons maintenant accès à :

- un fichier /etc/nsswitch.conf à mettre à jour ;
- un fichier /etc/libnss-ldap.conf à personnaliser ;
- une librairie libnss-ldap.so.

libnss - configuration

/etc/nsswitch.conf contient la liste des infos gérées par nss qu'il faut modifier :

```
/etc/nsswitch.conf
...
passwd:      compat ldap
group:   compat ldap
shadow:      compat ldap
hosts:    files dns
networks:    files
...
```

L'ajout de la mention *ldap* indique à la librairie nss qu'il y a un annuaire ldap qui peut nous fournir les données en plus des fichiers plats (sous debian, *compat* est un synonyme de *files*, cf le man). Attention à l'ordre des deux directives pour **d'abord** utiliser les fichiers plats (ça permet de conserver un accès root sur la machine, en cas de pb).
/etc/libnss-ldap.conf va permettre d'indiquer où est l'annuaire et comment s'y connecter. Le fichier doit être en lecture pour tous les utilisateurs :

```
mv2-debian:/etc# ls libnss* -la
-rw-r--r-- 1 root root 9809 2007-07-21 20:25 libnss-ldap.conf
-rw------- 1 root root   9 2007-07-21 20:25 libnss-ldap.secret
```

et il doit spécifier le dn à utiliser pour se connecter à l'annuaire, dn qui doit désigner un objet ayant le droit de lire l'attribut userPassword (le bind anonyme pose pb dans la config par défaut, mais on reviendra sur cet aspect sécurité une autre fois !).

```
/etc/libnss-ldap.conf

# Your LDAP server...
host 192.168.1.102
# The distinguished name of the search base.
base dc=labocned,dc=fr
```

Tests

1. Utiliser la commande id sur un login qui n'existe que dans l'annuaire :

```
mv2-debian:/etc# id phochon
uid=1234(phochon) gid=600(utilisateurs) groupes=600(utilisateurs)
```

2. Il faut que notre utilisateur possède un répertoire personnel avec les droits adéquats :

```
mv2-debian:/etc/ldap# mkdir /home/phochon
mv2-debian:/etc/ldap# chown phochon /home/phochon/
mv2-debian:/etc/ldap# chgrp utilisateurs /home/phochon/
mv2-debian:/etc/ldap# ls -la /home
[...]
drwxr-xr-x    2 phochonutilisateurs      1024 2007-07-21 19:11 phochon
```

3. En interrogeant la base passwd nous voyons tous les utilisateurs (ldap ou pas) :

```
mv2-debian:/etc# getent passwd
```

Maintenant que tout marche, on va tenter un login avec un compte ldap.
Ouvrez un tty2 :

```
mv2-debian login : phochon
Password: <tapez votre mot de passé>

Login incorrect !!
```

Il nous reste une dernière configuration, le pam : dans /etc/pam.d se trouve les fichiers de configuration pour l'authentification et il faut ajouter ldap :

```
mv2-debian:/etc/pam.d# vi common-auth
auth        sufficient      pam_ldap.so nullok_secure
auth        required        pam_unix.so nullok_secure

mv2-debian:/etc/pam.d# vi common-account
account     sufficient      pam_ldap.so
account     required        pam_unix.so

mv2-debian:/etc/pam.d# vi common-password
password    sufficient      pam_ldap.so use_authok
password    required        pam_unix.so nullok...

Mv2-debian:/etc/pam.d# vi common-session
session     optional        pam_ldap.so
session     required        pam_unix.so
```

Paul HOCHON doit pouvoir ouvrir une session :

```
mv2-debian login: phochon
Password:
Last login: Sun Jul 22 12:28:25 2007 on tty2
Linux mv2-debian 2.6.8-3-386 #1 Thu Sep 7 05:39:52 UTC 2006 i686 GNU/Linux

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
1 échec depuis la dernière connexion.
Le dernier échec était le dim 22 jui 2007 12:36:05 CEST sur tty2.
phochon@mv2-debian:~$
```

• Utilisation de l'annuaire à partir de Windows

Eh oui quand même c'est l'objectif !
Si vous essayez maintenant :

ça ne marchera pas : le domaine labocned.fr est géré par Samba et l'authentification de phochon par ldap, mais les deux ne sont pas encore liés !
http://cj.tronquet.free.fr/doc/samba3ldap.php
http://contribs.martymac.com/OpenLDAPFormationEOF/Formation-OpenLDAP-EOF.pdf
Tout d'abord nous avons mentionné que LDAP fonctionne selon des objectClass lesquels sont organisés selon un schéma : nous allons incorporer le schéma samba à LDAP (le paquetage Samba-doc doit être installé)…

```
mv2-debian:/etc/ldap/schema# cp /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz
```

Puis le décompresser :

```
mv2-debian:/etc/ldap/schema# gunzip /etc/ldap/samba.schema.gz
```

Modifiez /etc/ldap/slap.conf pour ajouter le schéma samba :

```
# Schema and objectClass definitions
include     /etc/ldap/schema/core.schema
include     /etc/ldap/schema/cosine.schema
include     /etc/ldap/schema/nis.schema
include     /etc/ldap/schema/inetorgperson.schema
include     /etc/ldap/schema/samba.schema
```

Redémarrez LDAP :

```
mv2-debian:/etc/ldap# /etc/init.d/slapd restart
```

Passons à la configuration de Samba (/etc/samba/smb.conf) :

```
mv2-debian:/etc/ldap# vi /etc/samba/smb.conf
```

Nous allons remplacer la base des pass par celle de ldap.
Changez :

```
passdb backend = tdbsam
```

En :

```
passdb backend = ldapsam : ldap://localhost
ldap admin dn = "cn=admin,dc=labocned,dc=fr"
ldap ssl = off
ldap delete dn = no
ldap user suffix = ou=utilisateurs
ldap machine suffix = ou=machines
ldap group suffix = ou=groupes
ldap suffix = dc=labocned,dc=fr
```

Indiquez à Samba le mot de passe à la base LDAP :

```
mv2-debian:/etc/ldap# smbpasswd –w <votre mot de passe ldap>
```

Redémarrez Samba :

```
mv2-debian:/etc/ldap# /etc/init.d/samba restart
```

Maintenant la création d'utilisateurs sous Samba les inclura dans LDAP.
Pour approfondir nous vous conseillons le paquet **smbldap-tools**…

• Quelques commandes de plus

Vous allez faire maintenant quelques manipulations en autonomie. Aidez-vous de l'aide en ligne ou de toute autre source.

Exercice 8

*L'utilisateur HOCHON a changé de téléphone personnel qui est maintenant le 0323453235. En utilisant la commande **ldapmodify**, faites la modification. Contrôlez le résultat avec la commande **ldapsearch**.*

Exercice 9

*Ajoutez à Paul HOCHON, l'attribut **title** avec la valeur « Enseignant ». Contrôlez le résultat avec la commande **ldapsearch**.*

Exercice 10

*Avec **ldapdelete**, supprimez l'utilisateur Paul HOCHON.*

Dépannage

Faites très attention aux espaces, en particulier en fin de ligne ! **Il ne doit y en avoir aucun**. Cela m'a causé pas mal de déboires.
Si vous voulez repartir sur une base LDAP « vide », arrêtez le démon ldap, détruisez les fichiers (/usr/lib/ldap), puis redémarrez le serveur.

Résumé

Un annuaire LDAP est un endroit central où sont stockés et organisés des objets du système informatique de l'entreprise.
La structure d'un annuaire LDAP est hiérarchique. La racine est généralement représentée par le nom de domaine de l'entreprise. Les unités d'organisation sont des sortes de dossiers dans lesquels on place des objets. Les objets peuvent être des utilisateurs[1], des contacts, des appareils ou des logiciels. Chaque catégorie d'objet est définie par une classe. Chaque classe permet de savoir quels attributs peuvent être associés à l'objet.
L'annuaire peut ensuite servir de référence à de nombreuses applications de l'entreprise : messagerie, identification des utilisateurs (login, samba, web, ftp, etc.).

Si vous voulez approfondir

Consultez les pages http://www.commentcamarche.net/ldap pour plus de détails. Elles sont très bien construites.
Cherchez comment utiliser LDAP à partir de Samba, Apache, ProFTPD, etc.
Et pour tout savoir, lisez « LDAP ADMINISTRATION SYSTEME » de CARTER publié chez O'REILLy France (ISBN : 2-84177-293-4).

[1] *Je ne cherche pas à dire que les humains sont des objets !!!*

Séquence 4

Un peu de théorie sur LDAP

Durée indicative : 1 heure

Cette séquence explique le fonctionnement général d'un annuaire LDAP.

▶ *Prérequis*

- **Avoir réalisé l'atelier de mise en œuvre d'un annuaire LDAP.**

▶ *Contenu*

1. **Rappel** 59

2. **Organisation d'un annuaire LDAP** 60

3. **Les objets LDAP** 61

4. **Les commandes LDAP** 62

5. **Notion d'héritage** 63

▶ *Capacités attendues*

- **Être capable d'évoquer ces notions théoriques lors de manipulations pratiques d'un annuaire LDAP.**

1. Rappel

Pacôme vous a expliqué quelques concepts :

- qu'est-ce qu'un annuaire ?
- à quoi cela sert ? etc.

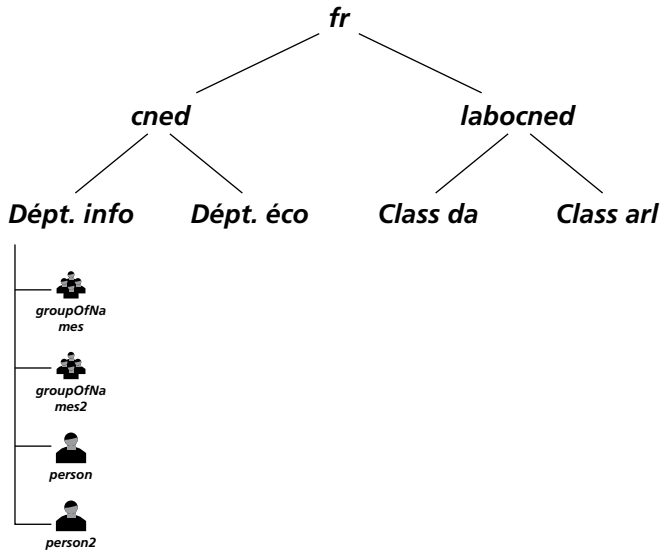
Nous allons ici entrer un petit peu plus dans le détail, tout en restant raisonnable…

Donc LDAP signifie ***Lightweight Directory Access Protocol,*** autrement dit c'est un protocole léger d'accès à un annuaire. L'annuaire géré est lui à la norme X500.
LDAP a été développé en 1995 par une équipe de l'université du Michigan pour remplacer le protocole d'accès DAP jugé trop lourd et donc peu performant.

Un annuaire sert donc à gérer hiérarchiquement des objets (eh oui parfois même un utilisateur doit être considéré comme un objet). Un annuaire est donc une base de données. Mais c'est une base de données un peu particulière : il est très performant en lecture et un peu moins en écriture… C'est ainsi qu'il sera capable très rapidement de trouver un objet dans l'organisation et d'en retourner les attributs, mais il mettra un peu plus de temps pour insérer de nouveaux objets.

2. Organisation d'un annuaire LDAP

LDAP est organisé suivant une structure arborescente hiérarchique, un peu comme le système de gestion de fichier de Linux :



L'arbre est appelé DIT : ***Directory Information Tree***. Il commence par une **racine**.
Puis se greffent les différentes branches, et à chaque **nœud** se trouve une **DES** : ***Directory Service Entry, une entrée de l'annuaire***.
Chaque entrée correspond à un **objet** : on parle d'objet car cela peut correspondre certes à une personne, mais aussi à un groupe de personnes, une imprimante, etc.
Comme un annuaire est une base de données, chaque objet est caractérisé par des **attributs** qui peuvent prendre une ou plusieurs valeurs.
Comme pour une base de données, l'attribut le plus significatif est **l'OID** : ***Object Identifier***.
Le **schéma** est l'ensemble des caractéristiques de tous les objets : description de leurs classes, de leurs attributs, etc.
Nous venons de présenter à quoi ressemble un annuaire LDAP et de définir un vocabulaire, entrons un peu plus dans le détail…

3. Les objets LDAP

3A. *Les classes d'objets LDAP*

Un objet LDAP appartient à une ou plusieurs classes d'objets.

Ces classes définissent les attributs obligatoires et optionnels qui décrivent l'objet, en voici une liste exemple :

Classe d'objets	Attributs obligatoires	Attributs optionnels
OrganizationalUnit (Unité Organisationnelle)	• OU • ObjectClass	• BusinessCategory • Description • FacsimileTelephoneNumber • Location • PostalAdress • SeeAlso • TelephoneNumber
InetOrgPerson (Personne de l'organisation)	• CommonName • Surname • ObjectClass	• BusinessCategory • CarLicense • DepartmentNumber • Description • EmployeeNumber • FacsimileTelephone • Number • GivenName • Mail • Manager • Mobile • OrganisationalUnit • Pager • PostalAdress • RommNumber • Secretary • SeeAlso • TelephoneNumber • Title • LabeledURI • UID

3B. Les attributs

Chaque entrée de l'annuaire correspond donc à un enregistrement décrit par une suite d'attributs, exemple : CommonName est un attribut.

Chaque attribut est décrit par :

- un nom identifiant : CommonName par exemple ;
- un OID : identifiant d'objet ;
- s'il peut prendre une ou plusieurs valeurs : on dira qu'il est mono ou multi-valué ;
- une syntaxe et des règles ;
- un indicateur d'usage ;
- un format ou une limite de taille.

3C. Le DN : Distinguished Name

Dans l'arborescence chaque entrée peut être repérée grâce au **DN** : ***Distinguished Name***.

Cela correspond au chemin dans l'arborescence qui conduit à l'objet, dans notre exemple le DN de person serait :

uid=person, ou=dept info, dc=cned, dc=fr

uid : user identifier

ou : organizational unit

dc : domain component

4. Les commandes LDAP

LDAP nous met à disposition un certain nombre de commandes qui permettent d'accéder et de modifier : on s'en sert comme des commandes de manipulation de fichier (cp, mv…)

Commande LDAP	Description
Search	**Recherche dans l'annuaire des objets**
Compare	**Comparaison de deux objets**
Add	**Ajouter une entrée**
Modify	**Modifier une entrée**
Delete	**Suppression d'un objet**
Rename	**Modifier le DN d'une entrée**
Bind	**Se connecter au serveur**
Unbind	**Se déconnecter**
Abandon	**Abandon d'une commande en cours**

5. Notion d'héritage

Un objet qui, dans l'arborescence, dépend d'un autre objet est dit objet « fils », exemple : person dépend de dépt info = dépt info est le « père » et person est le « fils »…

Comme chez le notaire le fils hérite des caractéristiques liées aux attributs de son père : c'est ce que l'on appelle **héritage**.

De même un objet peut appartenir à plusieurs classes d'objets : ses attributs sont alors l'ensemble des attributs des différentes classes.

Bien, nous avons expliqué l'essentiel, en espérant ne pas avoir été trop fastidieux…

En cas de besoin vous pouvez vous référer au site archinet :

http://www.linux-france.org/prj/edu/archinet/systeme/index_monopage.html

Gestion des disques

Durée approximative de cet atelier : 3 heures

▶ *Objectif*

Détailler les commandes qui permettent de gérer le système de fichiers de Linux : installer, partitionner, formater, attribuer des quotas et mettre des disques en miroir (RAID).

▶ *Conditions préalables*

Aucune en particulier.

▶ *Considérations techniques*

La partie sur le RAID concerne le RAID logiciel. De plus en plus de cartes mères de PC intègrent cette fonctionnalité mais la gèrent au niveau du contrôleur de disque, donc au niveau matériel : c'est indépendant du système d'exploitation.

Introduction

Un serveur représente souvent une unité de stockage importante où se trouve la plupart des données de l'entreprise. Il arrive régulièrement que cette capacité arrive à saturation. Nous allons voir comment ajouter et mettre en œuvre de nouveaux disques.
Ces données doivent aussi être protégées. Nous allons voir comment mettre en place un système RAID logiciel.

Installation d'un nouveau disque

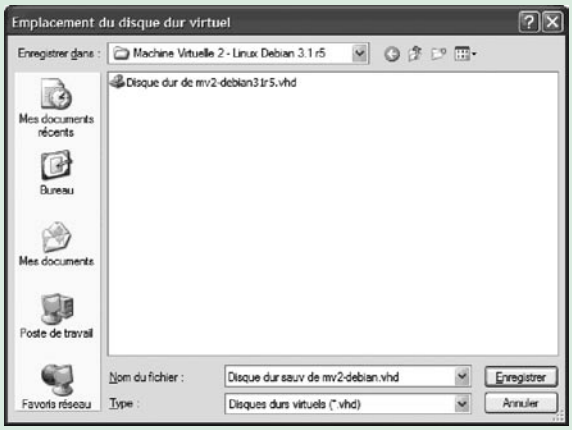
Commençons par un petit exercice de révision qui nous renvoie aux tous premiers ateliers.

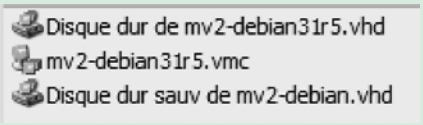
Exercice 11

Comment se nomment les disques sous Linux ? Comment se nommerait la deuxième partition du premier disque IDE ?

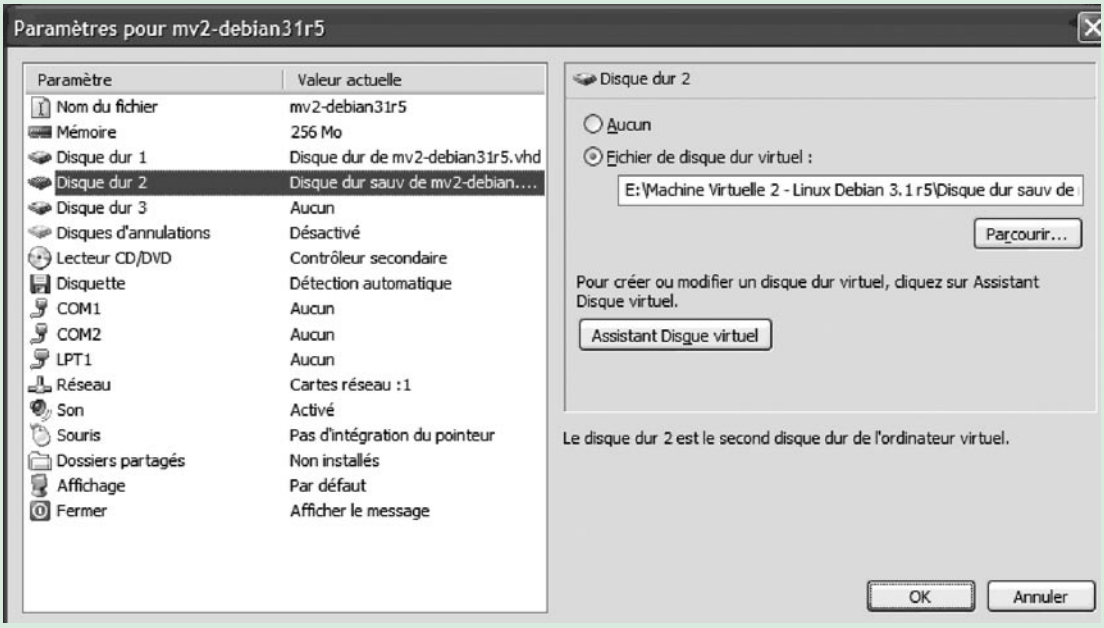
• Création du nouveau disque virtuel

Je vous invite bien sûr à lire la documentation de Virtual PC 2004 qui est fournie et stockée dans votre répertoire d'installation/Documentation/.
Dans Virtual PC, sélectionnez votre mv2-debian clic droit paramètres, disque virtuel n° 2, choisissez Assistant disque dur virtuel et créez un nouveau disque virtuel que vous stockerez dans le répertoire de mv2-debian :





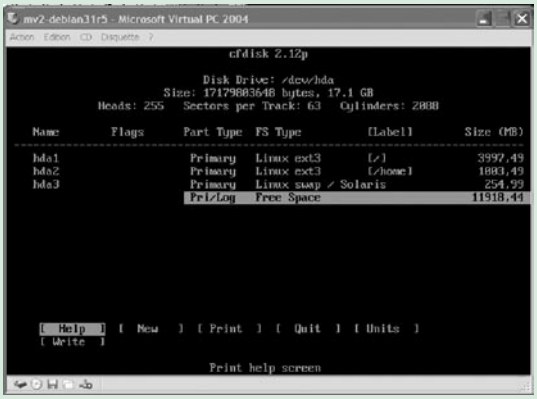
Vous venez d'installer un nouveau disque dur dans la machine.



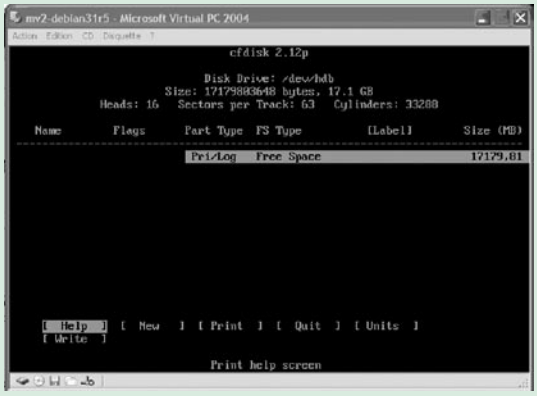
Redémarrez la machine virtuelle…
http://www.andesi.org/index.php?node=67

Connectez-vous en root et tapez :

```
mv2-debian:/etc# cfdisk </dev/hd+lettre a ou b ou...>
```


mv2-debian31r5 - Microsoft Virtual PC 2004
cfdisk 2.12p
Disk Drive: /dev/hda
Size: 17179803648 bytes, 17.1 GB
Heads: 255 Sectors per Track: 63 Cylinders: 2088
Name Flags Part Type FS Type [Label] Size (MB)
hda1 Primary Linux ext3 [/] 3997,49
hda2 Primary Linux ext3 [/home] 1883,49
hda3 Primary Linux swap / Solaris 254,99
Pri/Log Free Space 11918,44
[Help] [New] [Print] [Quit] [Units]
[Write]
Print help screen

/dev/hda notre premier disque…

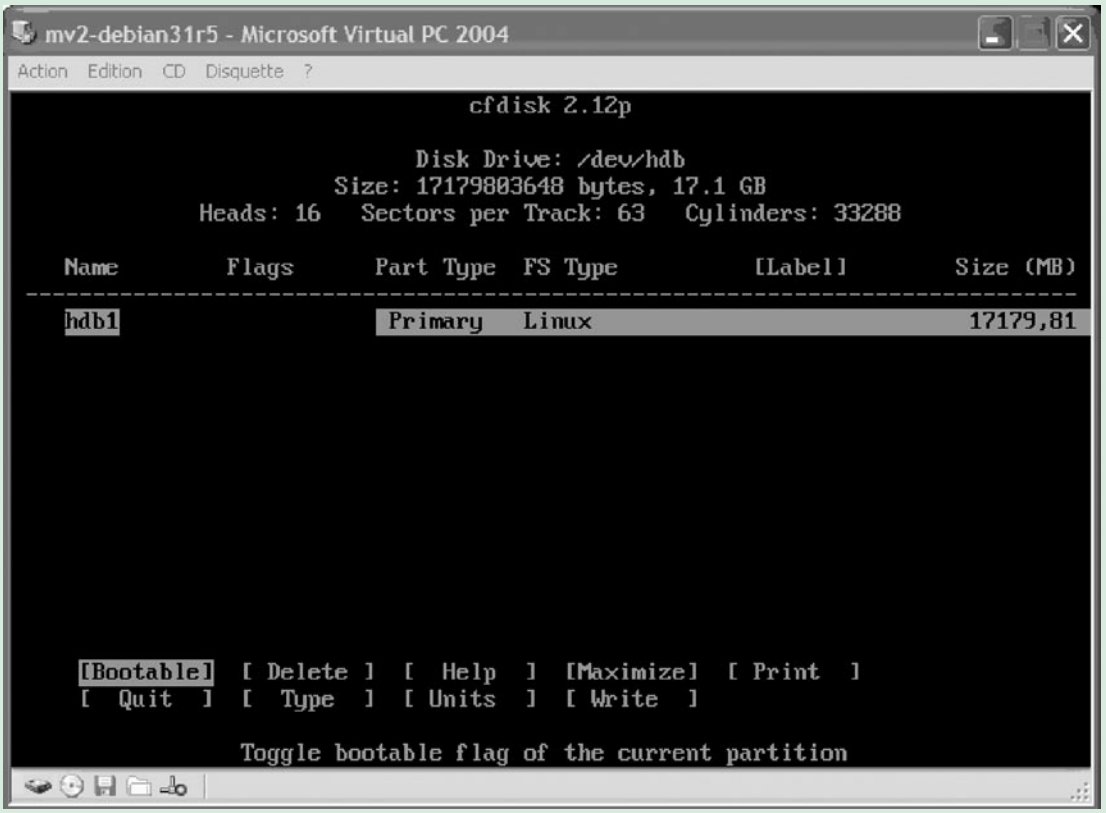

mv2-debian31r5 - Microsoft Virtual PC 2004
cfdisk 2.12p
Disk Drive: /dev/hdb
Size: 17179803648 bytes, 17.1 GB
Heads: 16 Sectors per Track: 63 Cylinders: 33288
Name Flags Part Type FS Type [Label] Size (MB)
Pri/Log Free Space 17179,81
[Help] [New] [Print] [Quit] [Units]
[Write]
Print help screen

/dev/hdb voilà notre nouveau disque !

Le logiciel rappelle à mon bon souvenir que le nouveau disque ne contient aucune partition : c'est pour le moment un espace libre. Qu'à cela ne tienne ! Nous allons en créer une.

• Partitionnement

Dans l'exemple, nous allons créer une partition primaire de toute la capacité du disque :


mv2-debian31r5 - Microsoft Virtual PC 2004
Action Edition CD Disquette ?
cfdisk 2.12p
Disk Drive: /dev/hdb
Size: 17179803648 bytes, 17.1 GB
Heads: 16 Sectors per Track: 63 Cylinders: 33288
Name Flags Part Type FS Type [Label] Size (MB)
hdb1 Primary Linux 17179,81
[Bootable] [Delete] [Help] [Maximize] [Print]
[Quit] [Type] [Units] [Write]
Toggle bootable flag of the current partition

N'oubliez pas **[Write]** à la fin pour écrire les données de la table de partition du disque ! À ce stade, la nouvelle partition a été enregistrée sur le disque. Il faut passer maintenant au formatage.

• Formatage

Je ne vous apprendrai rien en vous disant que chaque système d'exploitation possède son propre système de fichiers.
Le système de fichiers de prédilection de Linux est *ext3*.

```
mv2-debian:/# mke2fs –j /dev/hdb1
mke2fs 1.40-WIP (14-Nov-2006)
Etiquette de système de fichiers=
Type de système d'exploitation : Linux
Taille de bloc=4096 (log=2)
Taille de fragment=4096 (log=2)
2097152 i-noeuds, 4194280 blocs
209714 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=0
Nombre maximum de blocs du système de fichiers=0
128 groupes de blocs
32768 blocs par groupe, 32768 fragments par groupe
16384 i-noeuds par groupe
Superblocs de secours stockés sur les blocs:
   32768, 98304, 163840, 229376, 294912, 819200, 884736 etc.

Écriture des tables d'i-noeuds : complété
Création du journal (32768 blocs) : complété
Écriture des superblocs et de l'information de compatibilité du système de fichiers : complété

Le système de fichiers sera automatiquement vérifié tous les 22 montages ou après 180 jours, selon
la première éventualité. Utiliser tune2fs –c ou –i pour écraser la valeur.
```

Tout est ok.

• Montage

Vous savez que pour avoir accès à une unité de disque, il faut la monter :

```
mv2-debian:/# mkdir /mnt/disque2
mv2-debian:/# mount /dev/hdb1 /mnt/disque2
kjournald starting. Commit interval 5 seconds
EXT3 FS on hdb1, internal journal
EXT3-fs: mounted filesystem with ordered data mode.
mv2-debian:/# ls /mnt/disque2 -la
total 24
drwxr-xr-x      3 root    root        4096    2007-07-22 16:31.
drwxr-xr-x      3 root    root        1024    2007-07-22 16:44..
drwx------      2 root    root        16384   2007-07-22 16:31 lost+found
```

C'est bien joli, mais je n'ai pas envie de taper ces commandes à chaque fois ! J'aimerais que le disque soit monté automatiquement à chaque démarrage.

Modifiez le fichier **/etc/fstab** ainsi :

```
mv2-debian:/# cat /etc/fstab
# /etc/fstab : static file system information.
#
# <file system>  <mount point>  <type>  <options>  <dump>  <pass>
proc             /proc          proc    defaults   0       0
/dev/hda1        /              ext3    defaults   0       1
/dev/hda2        /home          ext3    defaults   0       0
/dev/hda3        none           swap    sw         0       0
/dev/hdb1        /mnt/disque2   ext3    defaults   0       0
```

La dernière ligne a été ajoutée. Pour le détail des options, regardez man fstab.

• Test d'une partition

Il faut d'abord démonter la partition :

```
mv2-debian:/# umount /mnt/disque2/
mv2-debian:/# fsck /dev/hdb1
fsck 1.40-WIP (14-Nov-2006)
e2fsck 1.40-WIP (14-Nov-2006)
/dev/hdb1: clean, 11/2097152 files, 109875/4194280 blocks
```

Le disque est *clean*, donc en parfait état de fonctionnement.

Activation du RAID

Le RAID1 est une sécurité supplémentaire sur les disques durs. On perd de la place mais si un disque a un problème, le ou les autres conservent les données et prennent le relais. Pour plus de détails sur le RAID et les différents niveaux, consultez votre cours d'AMSI de première année.
Il existe donc différents niveaux mais dans cet atelier, nous allons mettre en place du RAID❶ (*mirroring*) qui duplique les données sur les disques spécifiés.

Avantages :

- haute disponibilité des données et du système ;
- gain de performance en lecture.

Inconvénients :

- perte de temps CPU ;
- perte de vitesse en écriture ;
- perte d'espace disque.

• Définition des disques en RAID

http://www.debian-administration.org/articles/238

Installation des paquetages :

```
mv2-debian:/# apt-get install mdadm
```

mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de mdadm
Si le système de fichiers racine se trouve sur un ensemble MD (RAID), il doit être lancé au début de la procédure de démarrage. Si le système de fichiers racine se trouve sur un volume logique (« LVM »), qui se trouve aussi sur un volume MD, tous les composants de l'ensemble doivent être démarrés.
Si vous savez exactement quels sont les ensembles RAID nécessaires au démarrage du système de fichiers racine et si vous souhaitez différer le démarrage de tous les autres ensembles, veuillez les indiquer ici. Vous pouvez aussi indiquer « all » pour démarrer tous les ensembles existants.
Si vous n'avez pas besoin ou ne souhaitez pas démarrer d'ensemble RAID pour le système de fichiers racine, veuillez laissez l'entrée vide (ou indiquez « none »). Ceci peut être le cas si vous utilisez l'option
<Ok>

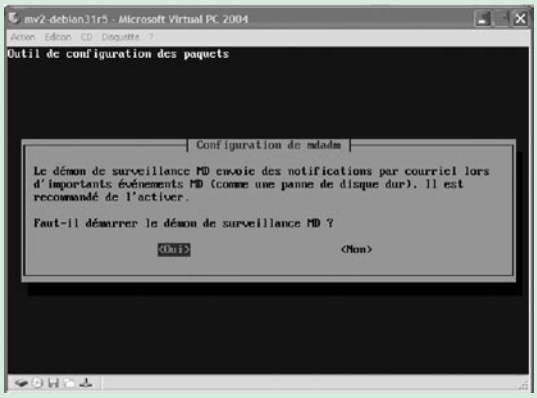
mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de mdadm
Ensembles MD requis par le système de fichiers racine :
all
<Ok>

laissez all

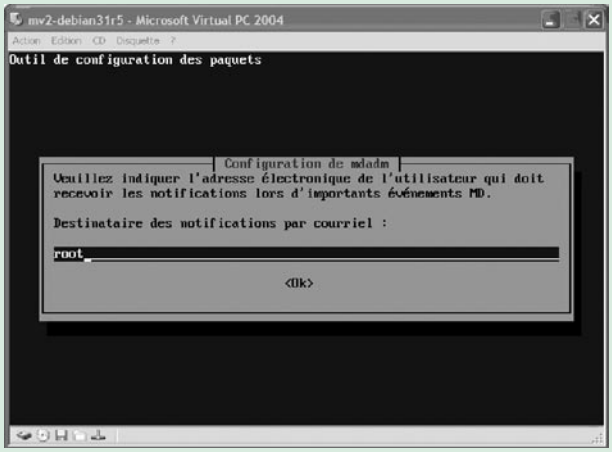
❶ *Redundant Array of Inexpensive Disks*

mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de mdadm
Si le noyau le gère (à partir de la version 2.6.14), mdadm peut vérifier périodiquement la redondance des périphériques RAID. Cette action peut demander beaucoup de ressources selon la configuration, mais cela aide à prévenir les rares cas de pertes de données. Notez que ce test est réalisé en lecture seule à moins que des erreurs ne soient rencontrées. Si des erreurs sont détectées, mdadm essaiera de les corriger, ce qui entraînera des écritures sur le média.
Par défaut, la vérification s'effectuera tous les premiers dimanche du mois à 01 h 06.
Faut-il vérifier chque mois la redondance des ensembles RAID ?
<Oui>
<Non>

<Oui> car le contrôle de redondance cyclique, comme vous l'avez vu, permet de vérifier l'intégrité des données.


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de mdadm
Le démon de surveillance MD envoie des notifications par courriel lors d'importants événements MD (comme une panne de disque dur). Il est recommandé de l'activer.
Faut-il démarrer le démon de surveillance MD ?
<Oui>
<Non>

<Oui> (nous aurons certainement un message d'erreur car nous n'avons pas encore de RAID à monitorer…)


mv2-debian31r5 - Microsoft Virtual PC 2004
Outil de configuration des paquets
Configuration de mdadm
Veuillez indiquer l'adresse électronique de l'utilisateur qui doit recevoir les notifications lors d'importants événements MD.
Destinataire des notifications par courriel :
root
<Ok>

Root sera averti par courriel des notifications

• Changement du type de partition

Pour les deux disques, vous allez devoir changer le type de partition qui est classiquement 83 (ext2/ext3) en fd (Auto-détection RAID Linux). Ce que vous pouvez visualiser sur le disque original hda :

```
mv2-debian:/# sfdisk –l /dev/hda
Disk /dev/hda : 33288 cylinders, 16 heads, 63 sectors/track
   Device Boot Start End #cyls #blocks Id System
/dev/hda1   0+    485   486-  3903763+ 83  Linux
/dev/hda2   486   607   122   979965    83  Linux
/dev/hda3   608   638   31    249007+     82  Linux swap / Solaris et le reste Empty
```

Recopions la structure des partitions de hda sur hdb :

```
mv2-debian:/# sfdisk –d /dev/hda | sfdisk /dev/hdb
```

Nous allons changer le type 83 de hdb1 et hdb2 (pas la swap !) en type **fd** pour auto détection, vous travaillez avec la commande **fdisk** :

```
mv2-debian:/# fdisk /dev/hdb

Commande (m pour aide) : p # n'hésitez pas à taper m...

Commande (m pour aide) : t
Selected partition 1
Hex code (type L to list codes) : fd

Commande (m pour aide) : p

Disk /dev/hdb : 17.1 GB, 17179803648 bytes
16 heads, 63 sectors/track, 33288 cylinders
Units = cylinders sur 1008 * 512 = 516096 bytes

   Device Boot Start     End    Blocks     IdSystème
/dev/hdb1    1     33288 16777120+   fdDétection auto RAID Linux

Commande (m pour aide) : w
La table de partition a été modifiée !
```

Faire idem pour hdb2.

Activons la swap de hdb :

```
mv2-debian:/# mkswap /dev/hdb3
mv2-debian:/# swapon -a
```

Et redémarrez pour que les modifications soient prises en compte comme Linux vous l'a expliqué…

• Activation du RAID

Un groupe de disques RAID se monte ensuite comme un disque classique. Il sera vu comme un disque unique, le *mirroring* étant géré par le noyau. Nous devons donc créer un **md0** pour gérer le mirroir hda1/hdb1 et un **md1** pour hda2/hdb2 : **attention de noter missing pour hda afin de conserver les données !**

```
mv2-debian:/# mdadm --create /dev/md0 --level 1 --raid-devices=2 missing /dev/hdb1
md: bind<hdb1>
md: md0: raid array is not clean – starting background reconstruction
md: raid1 personality registered as nr 3
raid1: raid set md0 active with 1 out of 2 mirrors
mdadm: array /dev/md0 started.
mv2-debian:/# mdadm --create /dev/md1 --level 1 --raid-devices=2 missing /dev/hdb2
```

Indiquons le type de système de fichier de /dev/md0 et /dev/md1 :

```
mv2-debian:/# mkfs.ext3 /dev/md0
mv2-debian:/# mkfs.ext3 /dev/md1
```

• Montage des disques RAID

Si vous regardez le système :

```
mv2-debian:/# cat /etc/fstab
# /etc/fstab: static file system information
#
# <file system>  <mount point>  <type>  <options>                 <dump>  <pass>
proc             /proc          proc    defaults                  0       0
/dev/hda1        /              ext3    defaults,errors=remount-ro 0      1
/dev/hda2        /home          ext3    defaults                  0       2
[...]
```

Vous constatez que le disque hdb ou md0 n'est pas monté.
Nous allons monter les partitions de md0 dans l'arborescence et copier le contenu de chaque partition de hda dans la partition correspondante (eh oui il faut bien « initialiser » le mirroring…) :

```
mv2-debian:/# mount /dev/md0 /mnt
kjournal starting. Commit interval 5 seconds
EXT3 FS on md0, internal journal
EXT3-fs: mounted filesystem with ordered data mode.
mv2-debian:/# cp –dpRx / /mnt
mv2-debian:/# mount /dev/md1 /mnt/home
mv2-debian:/# cp –dpRx /home /mnt/home
```

• Modification du fichier /etc/fstab

Exercice 12

Il faut associer dans le fichier /etc/fstab, notre système RAID avec les répertoires et /home de l'arborescence Linux. Vous le faites tout seul ?

• Mise en place de grub

Vous savez maintenant qu'au démarrage il y a le chargeur grub. Si le disque hda venait à tomber en panne ??? Pour le moment les données ne seraient pas perdues puisqu'elles sont mirrorées sur le deuxième disque, mais le système ne pourrait pas démarrer sur ce disque de sauvegarde ! Il nous faut y copier le chargeur :

```
mv2-debian:/# grub-install /dev/hdb
mv2-debian:/# grub
grub> device (hd0) /dev/hdb
grub> root (hd0,0)
grub> setup (hd0)
grub> quit
```

Mise en place de quotas

Il est bien pratique de pouvoir contrôler l'espace qu'occupent vos utilisateurs sur votre système de fichiers. Certains peuvent en abuser, surtout depuis l'introduction d'Internet... Heureusement les quotas sont là !

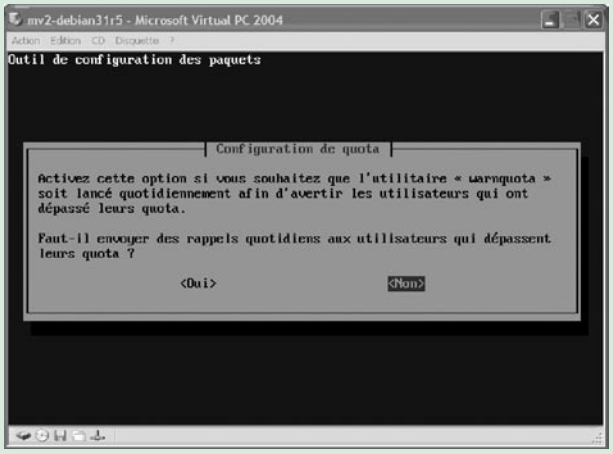
• Déclaration d'une partition avec quotas

http://www.debnet.info/procedure/linux/quotas.html

Nous allons introduire des quotas sur notre partition RAID[1] des comptes utilisateurs afin d'éviter que ce disque ne sature pas. Les utilisateurs qui franchiront une certaine limite seront prévenus et auront un certain délai pour régulariser.

```
mv2-debian:/# apt-get install quota
```

[1] *Si vous n'avez pas pu faire le RAID, vous pouvez quand même faire les quotas en adaptant les exemples.*


mv2-debian31r5 - Microsoft Virtual PC 2004
Action Edition CD Disquette ?
Outil de configuration des paquets
Configuration de quota
Activez cette option si vous souhaitez que l'utilitaire « warnquota » soit lancé quotidiennement afin d'avertir les utilisateurs qui ont dépassé leurs quota.
Faut-il envoyer des rappels quotidiens aux utilisateurs qui dépassent leurs quota ?
<Oui>
<Non>

Souhaitez-vous envoyer des rappels ? Là il s'agit d'un choix lié à la politique d'administration…

Dans le fichier /etc/fstab, ajoutez l'indication *usrquota* sur la partition md0 :

```
# /etc/fstab: static file system information
#
# <file system>  <mount point>  <type>  <options>                  <dump>  <pass>
   proc           /proc          proc    defaults                   0       0
[...]
  /dev/md0        /              ext3    defaults,errors=remount-ro 0       1
  /dev/md1        /home          ext3    defaults,usrquota          0       2
[...]
```

Ajoutez éventuellement *grpquota* si vous souhaitez gérer les quotas au niveau des groupes. Remontez le système de fichier :

```
mv2-debian:/# mount -o remount /home
```

Testez le fonctionnement des quotas :

```
mv2-debian:/# quotacheck /home -v
quotacheck: Scanning /dev/md1 [/home] done
```

• Attribution de quotas à un utilisateur

Pour attribuer un quota à un utilisateur, tapez la commande suivante en indiquant un utilisateur valide :

```
mv2-debian:/# edquota gilbert
Disk quotas for user gilbert (uid 1003):
 Filesystem   blocks  soft hard  inodes  soft hard
 /dev/md1     44      0    0     11      0    0
```

La première colonne correspond au nom du système de fichiers doté d'un quota activé. La deuxième colonne elle, affiche la place occupée (en Ko) actuellement par l'utilisateur. Les deux colonnes suivantes sont utilisées pour déterminer les limites douces (*soft limits*) et dures (*hard limits*).
Une limite dure correspond à la quantité maximale d'espace disque qu'un utilisateur ou groupe peut employer. Une fois cette limite atteinte, aucun espace supplémentaire ne peut être utilisé.
La limite douce définit la quantité maximale d'espace disque pouvant être utilisée. Néanmoins, contrairement à la limite dure, la limite douce peut être dépassée pendant un certain temps. On se réfère à cette durée sous le terme « période de grâce ». Cette dernière peut être exprimée en secondes, minutes, heures, jours, semaines ou mois. Elle sera paramétrée un peu plus loin.
La deuxième partie correspond aux *inodes*[1]. On peut établir des quotas sur le nombre de fichiers/répertoires et non sur leur volume. L'utilité de cette fonctionnalité reste à démontrer.
Activer les quotas : mv2-debian:/#quotaon-a

Exercice 13

Lancez la commande edquota sur votre utilisateur. Relevez l'espace occupé actuellement par ses fichiers. Introduisez une certaine limite douce (supérieure à cette valeur) puis une limite dure encore supérieure.
Maintenant, vous allez dépasser les limites ! Connectez-vous sous cet utilisateur, puis, dans son compte utilisateur avec une commande du style :

```
dd if=/dev/zero of=grosfichier count=<x> bs=<taille_en_octets>
```

vous allez chercher à atteindre la limite douce, puis la limite dure. Cette commande va créer un fichier rempli de zéros d'une taille de x fois taille_en_octets. À vous de trouver les bonnes valeurs.
En parallèle, en tant que *root* dans un autre terminal, vous étudierez l'évolution avec la commande repquota /home.

[1] *Comprenez les fichiers ou répertoires détenus par l'utilisateur.*

Résumé

L'installation physique d'un nouveau disque est suivie de son partitionnement (fdisk), de son formatage (mke2fs) et de son montage (dans /etc/fstab).

Le RAID est une technologie de sécurité appliquée aux disques. Linux sait mettre en œuvre du RAID logiciel. Le RAID1 ou *mirroring* duplique les données écrites sur un disque sur tous les autres inclus dans le groupe. La définition du groupe avec mdadm. Le type de partition des disques doit être passé en fd puis le groupe est créé avec la commande mdadm –create etc. Ensuite, le nouveau système de fichiers est géré comme un disque unique (on n'utilise plus les partitions individuelles mais le groupe référencé par /dev/mdx).

Le RAID ne remplace pas les sauvegardes !

Les quotas permettent de limiter automatiquement la place occupée par les fichiers des utilisateurs. Le fichier /etc/fstab doit être modifié afin d'indiquer sur quelles partitions on veut des quotas. Ensuite, la commande edquota permet de définir pour chaque utilisateur ou chaque groupe la quantité d'espace allouée.

Gestion des pannes (reprise)

Durée approximative de cet atelier : 2 heures

▶ *Objectif*

Faire des sauvegardes et étudier les principaux moyens de démarrer un système Linux en panne.

▶ *Conditions préalables*

Aucune en particulier.

▶ *Considérations techniques*

Posséder une unité de sauvegarde sur bande ou un graveur de disque pourrait être un bon complément à cet atelier.

Introduction

Il peut arriver que des problèmes divers, des manipulations, de mauvaises configurations, etc. empêchent de démarrer correctement le système. Il existe différents moyens de récupération que nous allons présenter dans cet atelier.

Mais avant tout, nous allons dire quelques mots au sujet des sauvegardes. Vous savez qu'elles sont indispensables dans un environnement informatique. L'apparente fiabilité du matériel cache des catastrophes mémorables. Ainsi, vous devriez sauvegarder régulièrement le système et les fichiers utilisateurs afin d'être en mesure de les restaurer au besoin. Un point important : le système RAID que nous avons mis en place à l'atelier précédent n'est **pas** une sauvegarde. Cela ne remplace pas la sauvegarde des fichiers sur un support externe conservé, si possible, hors de l'entreprise.

Dans tous les cas, il est très important de s'entraîner sur des machines de test, de simuler des pannes et de réaliser des restaurations afin d'être entraîné le jour j. On ne compte plus le nombre de personnes qui se croyaient à l'abri alors que leur système de sauvegarde ne fonctionnait pas ou mal.

Les sauvegardes

Une tâche classique d'un administrateur consiste à restaurer un système opérationnel que l'on avait sauvegardé avant une panne ou un problème. Nous allons maintenant voir comment sauvegarder et restaurer dans un environnement Unix.

• La commande tar

Pour faire une sauvegarde de votre partition système (ou de n'importe quel répertoire ou partition du système de fichiers), vous pouvez utiliser la commande *tar* puis graver le résultat sur un cédérom, l'envoyer sur un disque partagé sur le réseau ou sur une bande.

Prenons un exemple : nous allons archiver et compresser tous les fichiers de nos utilisateurs en les plaçant dans une archive nommée *sauve.tgz*. Elle sera placée dans le répertoire courant :

```
mv2-debian:~# tar -cpzPf sauve.tgz /home
```

Contrôlons :

```
mv2-debian:~# ls -la
[...]
-rw-r--r--   1 root    root    170952 2007-07-24 09:34 sauve.tgz
[...]
```

On peut lister les fichiers contenus dans l'archive et filtrer l'affichage. Par exemple, pour l'utilisateur gilbert :

```
mv2-debian:~# tar -tzf sauve.tgz | grep gilbert
[...]
home/gilbert/public_html/
home/gilbert/public_html/form.html
home/gilbert/public_html/index.html
home/gilbert/public_html/form.php
home/gilbert/public_html/liste.php
home/gilbert/.bash_history
home/gilbert/index.html
[...]
```

Dans le but de restaurer à partir de l'archive, supprimons un de ses fichiers dans son répertoire (bien sûr, vous ne faites pas les manipulations sur des utilisateurs réels !!!) :

```
mv2-debian:~# rm /home/gilbert/index.html
mv2-debian:~# ls /home/ gilbert/index.html
ls: /home/ gilbert/index.html: Aucun fichier ou répertoire de ce type
```

Maintenant, restaurons :

```
mv2-debian:~# tar -xzvPf /root/sauve.tgz /home/gilbert/index.html
/home/gilbert/index.html
mv2-debian:~# ls /home/ gilbert/index.html
/home/gilbert/index.html
```

On peut, bien sûr, restaurer tout le contenu de la sauvegarde. Par contre, on ne peut pas filtrer avec des jokers. Consultez l'aide en ligne de la commande *tar* pour connaître toutes les possibilités.

• Création d'images disque

L'utilitaire libre *partimage* est très intéressant pour cloner des partitions entières. Je vous laisse le découvrir :
http://www.partimage.org

Les modes de démarrage du système

Linux offre plusieurs possibilités de démarrer une machine en panne.

• Le mode « interactif »

Vous avez un service (un démon) Linux mal configuré qui vous bloque le démarrage ? Ce n'est pas un problème. Lors de la séquence de démarrage, Linux vous indique qu'il passe en mode interactif (juste avant la série de OK). Si vous appuyez sur la touche correspondant à la lettre I à ce moment-là, vous pourrez ensuite choisir les services à démarrer ou non.

• Le mode maintenance

Dans certaines conditions (perte du mot de passe *root*, services réseaux mal configurés, interface graphique mal configurée...), vous pouvez avoir besoin de démarrer le système dans un mode minimal afin de réaliser des opérations de maintenance ou de récupération « légères ».
Pour accéder à ce mode, vous devez l'indiquer au noyau Linux avant son chargement. Vous en avez la possibilité lorsque l'invite de démarrage s'affiche : **recovery mode.**

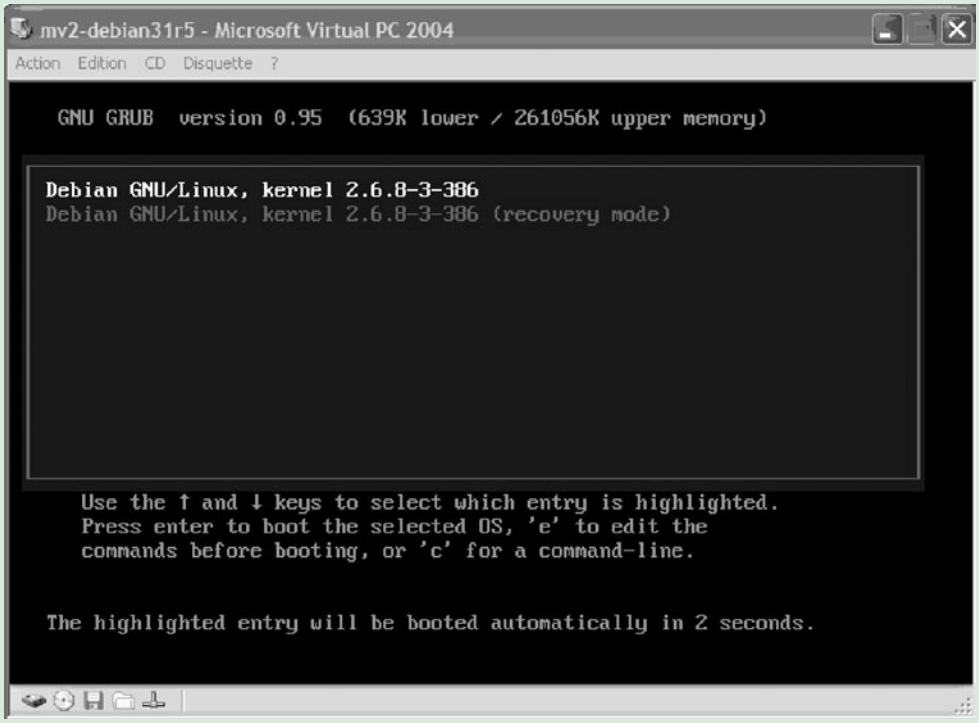

mv2-debian31r5 - Microsoft Virtual PC 2004
Action Edition CD Disquette ?
GNU GRUB version 0.95 (639K lower / 261056K upper memory)
Debian GNU/Linux, kernel 2.6.8-3-386
Debian GNU/Linux, kernel 2.6.8-3-386 (recovery mode)
Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
The highlighted entry will be booted automatically in 2 seconds.

Figure 1 : écran de démarrage de la machine

Sélectionnez le mode **recovery** avant la fin du décompte.

Exercice « libre » (pas de corrigé)

Testez ces différentes possibilités sur votre machine de test avant un réel problème afin d'être préparé !

• Le mode « récupération »

Votre système ne démarre plus du tout (par exemple, secteur de boot endommagé). Tout n'est peut-être pas perdu ! Vous pouvez essayer de démarrer à partir d'un support externe.

Disquette de démarrage

http://www.finix.eu.org/spip/article.php3?id_article=69
http://p.karatchentzeff.free.fr/freesoft/debian/debian-guide/html/ch4.html
Pour générer une disquette de démarrage, utilisez la commande *mkbootdisk* suivi de la version de votre noyau.

Pour connaître la version de son noyau :

```
mv2-debian:~# uname –r
2.6.8-3-386
```

Premièrement, formatez la disquette au format ext2fs et créez-y les répertoires boot et boot/grub :

```
# mke2fs /dev/fd0
# mkdir/floppy
# mount -t ext2 /dev/fd0 /floppy
# grub-install--root-directory=/floppy/dev/fd0
```

Pour copier le noyau sur la disquette :

```
# cp bzImage /floppy/boot
```

À ce stade, il n'y a plus qu'à créer un fichier : le fichier de configuration /floppy/boot/grub/menu.lst.
Le contenu suivant fera très bien l'affaire :

```
timeout 2
default 0
title Linux 2.6.8-3-386 (disquette)
root (fd0)
kernel /boot/bzImage root=/dev/hda1
# umount /floppy
```

Vous pouvez maintenant redémarrer sur la disquette. Si votre configuration matérielle est un peu complexe (disque SCSI par exemple) cela ne fonctionnera pas. Vous pouvez démarrer avec le CDROM d'installation ou une distribution spéciale récupération (voir ci-dessous).

Les supports disquette du CD-Rom

http://people.debian.org/~cts/debian-m68k/boot-floppies/current/READ-fr.txt

Une distribution spéciale récupération

Voyez à cette adresse, la possibilité de se créer un cédérom avec plein d'outils de récupération bien utiles : http://www.sysresccd.org/index.en.php

Résumé

Faire des sauvegardes est primordial. Énormément de solutions existent, à vous de trouver celle adaptée à vos besoins. *tar* est la commande « historique » présente sur tous les systèmes UNIX.
Différents moyens de redémarrer un système en panne existent. Si le système démarre mais se bloque ou vous empêche d'arriver au *login*, vous pouvez utiliser les modes de maintenance. **Attention**, comme vous l'avez constaté, ces modes ne demandent pas le mot de passe de *root*. Des petits malins peuvent démarrer votre système et réaliser des tâches qui nécessitent habituellement les autorisations de *root*. Un système Linux doit donc être protégé des accès physiques de personnes non habilitées.

Si vous voulez approfondir

Sur le chapitre des sauvegardes, il y a énormément de choses à dire :

- utiliser partimage ;
- voir du côté de la commande *rsync* pour gérer une copie à jour en permanence des fichiers situés dans un répertoire mais sur une autre machine ;
- programmer les sauvegardes avec les démons CRON et ANACRON.

Les scripts shell

Durée approximative de cet atelier : 3 heures

▶ *Objectif*

Découvrir les puissantes possibilités proposées à l'administrateur par le langage de shell. Vous allez créer des scripts en langage de shell afin d'automatiser des tâches d'administrations.

▶ *Conditions préalables*

Aucune en particulier.

▶ *Considérations techniques*

Cet atelier va faire appel à vos connaissances acquises tout au long de ce cours. Mais aussi à des compétences issues de votre cours d'algorithmique et de programmation. Je suppose donc dans cet atelier que vous maîtrisez les bases élémentaires de la programmation. Le langage de shell n'est pas un véritable langage de programmation, il est moins sophistiqué. L'intérêt est de permettre le traitement de lots de commandes shell en y ajoutant éventuellement des conditions. Toutefois, vous devrez savoir ce qu'est une variable, une boucle et une condition. Je suppose qu'à ce stade de votre formation, ce n'est pas un problème.

Introduction

Jusqu'à présent, nous avons saisi beaucoup de commandes *shell* « une par une ». Il est possible de regrouper des blocs de commandes dans un fichier de script. Un script bash est un fichier texte contenant une suite de commandes shell, exécutables par un interpréteur (ici le programme /bin/bash).

Le langage bash gère notamment :

- la gestion des entrées-sorties et de leur redirection ;
- des variables définies par le programmeur et des variables systèmes ;
- le passage de paramètres ;
- des structures conditionnelles et itératives.

Un script peut être lancé en ligne de commande ou bien dans un autre script.

Mes premiers scripts

Nous allons balayer les principales caractéristiques des scripts.

• Afficher un message

Un script étant un fichier texte, on doit l'éditer avec un outil comme **vi**. Tout script devrait commencer par la ligne :

```
#!/bin/bash
```

C'est obligatoire dans le cas où plusieurs shells sont installés sur la machine. En effet, il en existe plusieurs (sh, csh, ksh zsh, etc.) et chacun possède son propre langage. Dans cet atelier, nous nous contenterons de celui installé par défaut.
Ensuite, notre script doit comporter au moins une instruction. Par exemple, utilisons la commande **echo** qui permet d'afficher un message :

```
echo "Bonjour, le monde"
```

Donc, notre premier script aura cette tête-là :

```
#!/bin/bash
echo "Bonjour, le monde"
```

Éditez ce script avec vi, sauvegardez-le sous le nom script1.sh.
Listez le contenu :

```
mv2-debian:~# ls –la
[...]
-rw-r—r-- 1 root root 36 2007-07-24 10:30 script1.sh
```

Ensuite, il faut lui attribuer les permissions d'exécution :

```
mv2-debian:~# chmod u+x script1.sh
```

Maintenant, essayons de l'exécuter :

```
mv2-debian:~# script1.sh
-bash: script1.sh: command not found
```

Impossible de l'exécuter alors que nous sommes dans le répertoire qui contient le fichier. Quelle est la raison ? En fait, une variable du shell contient les chemins dans lesquels il faut chercher les fichiers exécutables. Cette variable s'appelle PATH et il est possible de consulter son contenu de cette façon :

```
mv2-debian:~# echo $PATH
/usr/local/sbin:/usr/local/bin:usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
```

Rien n'indique le répertoire courant ! Dès lors, il y a plusieurs solutions :

- soit ajouter au chemin le « . » qui représente le répertoire courant :

```
mv2-debian:~# PATH=$PATH:.
```

- soit, spécifier devant le script le répertoire courant :

```
mv2-debian:~#./script1.sh
Bonjour, le monde
```

• Récupérer des paramètres

On peut passer au script une liste de paramètres en les indiquant à la suite du nom du script et en les séparant par des espaces. Les valeurs de ces paramètres sont récupérables via des variables $1, $2, etc. Mais, contrairement aux langages de programmation classiques, ils ne peuvent pas être modifiés.
Par exemple, on peut faire le script2.sh :

```
#!/bin/bash
echo "Bonjour, $1 et bonne journée !"
```

Ce qui donne :

```
mv2-debian:~#./script2.sh toto
Bonjour, toto et bonne journée !
```

On peut également connaître le nombre de paramètres passé à un script avec la variable $#. Par exemple, script3.sh :

```
#!/bin/bash
echo "Vous avez saisi $# paramètres."
```

Ce qui donne :

```
mv2-debian:~#./script3.sh para1 para2
Vous avez saisi 2 paramètres.
mv2-debian:~#./script3.sh para1 para2 para3
Vous avez saisi 3 paramètres.
```

Enfin, sachez que la variable $0 représente le nom du script.

• Tester des conditions

Un script peut contenir des structures conditionnelles. Le modèle général est :

```
if [condition]
then
   action1
else
   action2
fi
```

Note : la place du **then** *et du* **else** *(seuls sur la ligne) est importante et* ***respectez les espaces.***

Par exemple, le script4.sh :

```
#!/bin/bash
if [ $# = 2 ]
then
   echo “Bonjour $2 $1 et bonne journée !”
else
   echo “Syntaxe : $0 nom prenom”
fi
```

Attention aux espaces !
Ce qui donne comme résultat :

```
mv2-debian:~#./script4.sh
Syntaxe :./script4.sh nom prenom
mv2-debian:~#./script4.sh HOCHON Paul
Bonjour Paul HOCHON et bonne journée !
```

De nombreuses possibilités de comparaison existent. Voici les principales :

Opérateur	Description	Exemple
Opérateurs sur des fichiers		
-e *filename*	**vrai si** *filename* **existe**	[-e /etc/shadow]
-d *filename*	**vrai si** *filename* **est un répertoire**	[-d /tmp/trash]
-f *filename*	**vrai si** *filename* **est un fichier ordinaire**	[-f /tmp/glop]
-L *filename*	**vrai si** *filename* **est un lien symbolique**	[-L /home]
-r *filename*	**vrai si** *filename* **est lisible (r)**	[-r /boot/vmlinuz]
-w *filename*	**vrai si** *filename* **est modifiable (w)**	[-w /var/log]
-x *filename*	**vrai si** *filename* **est exécutable (x)**	[-x /sbin/halt]
file1 **-nt** *file2*	**vrai si** *file1* **plus récent que** *file2*	[/tmp/foo -nt /tmp/bar]
file1 **-ot** *file2*	**vrai si** *file1* **plus ancien que** *file2*	[/tmp/foo -ot /tmp/bar]
Opérateurs sur les chaînes		
-z *chaine*	**vrai si la** *chaine* **est vide**	[-z «$VAR»]
-n *chaine*	**vrai si la** *chaine* **est non vide**	[-n «$VAR»]
chaine1 **=** *chaine2*	**vrai si les deux chaînes sont égales**	[«$VAR» = «totoro»]
chaine1 **!=** *chaine2*	**vrai si les deux chaînes sont différentes**	[«$VAR» != «tonari»]
Opérateurs de comparaison numérique		
num1 **-eq** *num2*	**égalité**	[$nombre -eq 27]
num1 **-ne** *num2*	**inégalité**	[$nombre -ne 27]
num1 **-lt** *num2*	**inférieur (<)**	[$nombre -lt 27]
num1 **-le** *num2*	**inférieur ou égal (< =)**	[$nombre -le 27]
num1 **-gt** *num2*	**supérieur (>)**	[$nombre -gt 27]
num1 **-ge** *num2*	**supérieur ou égal (> =)**	[$nombre -ge 27]

• Saisir une variable

Vous pouvez demander à l'utilisateur de saisir une variable. Par exemple, le script5.sh :

```
#!/bin/bash
echo “Donnez votre prénom et votre nom”
read prenom nom
echo “Bonjour $prenom $nom”
```

Ce qui donne :

```
mv2-debian:~#./script5.sh
Donnez votre prénom et votre nom
Paul HOCHON
Bonjour Paul HOCHON
```

On peut ensuite tester la saisie. Par exemple le script6.sh :

```
mv2-debian:~#$ cat script6.sh
#!/bin/bash
echo “Donnez votre prénom et votre nom”
read prenom nom
if [ “$prenom” = “” ]
then
   echo “Vous n’avez pas saisi de prénom !”
else
   echo “Bonjour $prenom $nom”
fi
```

Note : la variable doit impérativement être entourée de guillemets car juste avant l’exécution de la ligne, le shell remplace la variable par son contenu. Si on ne met pas les guillemets mais que cette variable contient des espaces, le shell sera un peu perdu et générera une erreur.

Le script6.sh donne :

```
mv2-debian:~#./script6.sh
Donnez votre prénom et votre nom

Vous n’avez pas saisi de prénom !
mv2-debian:~#./script6.sh
Donnez votre prénom et votre nom
Paul HOCHON
Bonjour Paul HOCHON
```

On aurait également pu l’écrire sous cette forme :

```
#!/bin/bash
echo “Donnez votre prénom et votre nom”
read prenom nom
if [ -z “$prenom” ]
then
   echo “Vous n’avez pas saisi de prénom !”
else
   echo “Bonjour $prenom $nom”
fi
```

• Faire des boucles

Le langage intègre différentes structures répétitives : le *for* (pour) et le *while* (tant que).

Le for

Lorsque l'on connaît d'avance le nombre d'itérations, on utilise généralement la structure for :

```
for variable [in liste]
do
      commandes (utilisant $variable)
done
```

Par exemple, le script7.sh :

```
#!/bin/bash
# Liste les planètes.

for planete in Mercure Vénus Terre Mars Jupiter Saturne Uranus Neptune Pluton
do
   echo $planete
done
```

Le résultat est le suivant :

```
mv2-debian:~#./script7.sh
Mercure
Vénus
Terre
Mars
Jupiter
Saturne
Uranus
Neptune
Pluton
```

Ce qui fait que pour compter jusqu'à 10, il faudrait rédiger un script8.sh de cette manière :

```
#!/bin/bash
# Syntaxe standard.

for a in 1 2 3 4 5 6 7 8 9 10
do
   echo -n "$a "
done
```

Bof. Ça marche, mais ce n'est pas terrible. Heureusement, bash accepte une syntaxe proche du langage C comme dans le script8.sh :

```
#!/bin/bash
# Syntaxe style C.

LIMITE=10  # initialisation de la variable

for ((a=1; a <= LIMITE ; a++)) # Double parenthèses et "LIMITE" sans "$".
do
      echo -n "$a "
done
```

Le while

Lorsque l'on ne connaît pas à l'avance le nombre d'itérations, on utilise une boucle *while*. La syntaxe générale est la suivante :

```
While [condition]
do
   commande...
done
```

Essayons le script9.sh :

```
#!/bin/bash

var0=0

while ["$var0" -lt 10]
do
   echo -n "$var0 "    # -n supprime le retour chariot.
   var0= $((var0 + 1)) # expr permet de faire un calcul var0=var0+1
done
```

Exercices

Cette partie n'est consacrée qu'à des exercices. J'espère vous montrer comment on peut progressivement construire un script. Pour vous aider, utilisez les documents indiqués dans la rubrique dépannage. Utilisez également les nombreuses sources disponibles sur internet. L'objectif est d'arriver à un résultat coûte que coûte.

Exercice 14

Écrivez un script qui affiche tout simplement le message « Vive Linux ». Résultat attendu :

```
mv2-debian:~#./tp.sh
Vive Linux
```

Exercice 15

Modifiez le script précédent afin d'afficher le message « Vive » suivi d'un texte passé en paramètre au script. Résultat attendu :

```
mv2-debian:~#./tp.sh
Vive
mv2-debian:~#./tp.sh toto
Vive toto
```

Exercice 16

Même question en contrôlant le nombre de paramètres. S'il est incorrect, on affiche un message d'erreur.

```
mv2-debian:~#./tp.sh
Vous devez fournir UN paramètre !
```

```
mv2-debian:~#./tp.sh toto titi
Vous devez fournir UN paramètre !
```

```
mv2-debian:~#./tp.sh toto
Vive toto
```

Exercice 17

Afficher les trois noms « riri, fifi et loulou » en utilisant impérativement une boucle *for*. Résultat attendu :

```
mv2-debian:~#./tp.sh
riri
fifi
lolou
```

Exercice 18

Demander à l'utilisateur de saisir un nom de fichier ou de répertoire. D'abord, le script vérifie s'il existe dans le répertoire courant. S'il existe, le script indique s'il s'agit d'un fichier ou d'un répertoire. Résultat attendu (en supposant que le répertoire tmp et le fichier script1.sh existent) :

```
mv2-debian:~#./tp.sh
Entrez un nom de fichier ou de répertoire
↵
Cela n'existe pas dans le répertoire courant !!!

mv2-debian:~#./tp.sh
Entrez un nom de fichier ou de répertoire
efo,jevhj
Cela n'existe pas dans le répertoire courant !!!

mv2-debian:~#./tp.sh
Entrez un nom de fichier ou de répertoire
tmp
C'est un répertoire

mv2-debian:~#./tp.sh
Entrez un nom de fichier ou de répertoire
script1.sh
C'est un fichier
```

Exercice 19

Vous transformez le script précédent de façon à ce qu'il parcoure tous les fichiers du répertoire courant et indique le type de chaque objet trouvé. Vous utiliserez la boucle suivante :

```
for nom in ./*
```

Le contenu du répertoire est le suivant :

```
mv2-debian:~# ls
script1.sh* script3.sh* script5.sh* script7.sh* script9.sh* tp.sh*
script2.sh* script4.sh* script6.sh* script8.sh* tmp/
```

Le résultat attendu est le suivant :

```
mv2-debian:~#./tp.sh
./script1.sh est un fichier
./script2.sh est un fichier
./script3.sh est un fichier
./script4.sh est un fichier
./script5.sh est un fichier
./script6.sh est un fichier
./script7.sh est un fichier
./script8.sh est un fichier
./script9.sh est un fichier
./tmp est un répertoire
./tp.sh est un fichier
```

Exercice 20

Vous faites un script qui balaye le contenu du fichier /etc/passwd et en affiche chaque ligne. Pour ce faire, vous utilisez une boucle sur le modèle suivant :

```
while read ligne
do
   vos_instructions...
done < /etc/passwd
```

Cette boucle utilise le principe de la redirection de fichier (représentée par le <). Le fichier est passé ligne par ligne à la boucle. Chaque ligne est récupérée dans une variable nommée ***ligne***.

Le résultat attendu est le suivant :

```
mv2-debian:~#./tp.sh
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
adm:x:3:4:adm:/var/adm:/bin/sh
[...]
```

Exercice 21

Vous savez que le fichier /etc/passwd est découpé en un certain nombre de champs. On vous demande de modifier le script précédent afin d'afficher le fichier d'une façon agréable à lire. Pour découper un champ d'une ligne, vous pouvez utiliser l'instruction suivante :

```
nom=$(echo «$ligne» | cut -d: -f1)
```

Expliquons cette instruction qui n'a rien de simple au premier abord :

1. L'instruction **echo "$ligne"** affiche le contenu de la variable ligne ;
2. mais en fait, il n'y aura pas d'affichage. Ce qui aurait dû être affiché est passé en paramètre à la commande cut (par le symbole |) ;
3. la commande cut est prévue pour découper une ligne en un certain nombre de champs. Il faut lui indiquer le caractère séparateur de champs (:) et le champ que l'on souhaite obtenir (ici, le premier). Consultez le man de cut ;
4. enfin, le résultat de tout ça (donc le premier champ extrait de la ligne) est placé dans une variable dénommée *nom*.

Le résultat attendu est :

```
mv2-debian:~#./tp.sh /etc/passwd
Le compte de l'utilisateur root est situé dans /root
Le compte de l'utilisateur bin est situé dans /bin
Le compte de l'utilisateur daemon est situé dans /sbin
[...]
```

Exercice 22

Vous avez un fichier de données contenant une liste d'utilisateurs avec leur mot de passe. Voici un exemple de ce fichier :

```
mv2-debian:~# cat users.txt
toto:passtoto
titi:passtiti
tata:passtata
```

Vous voulez automatiser la création d'utilisateurs. Vous proposerez un script qui prend en paramètre le nom du fichier de données, vérifie son existence puis qui crée les utilisateurs (commande useradd) et définit leur mot de passe (commande passwd).

Sachez que la commande exit -1 permet de quitter le script à tout moment, en renvoyant un code d'erreur.

Le résultat doit être le suivant :

```
mv2-debian:~#./tp.sh
Usage :./tp.sh <nom_de_fichier>
mv2-debian:~#./tp.sh houba
Le fichier de données n'existe pas !
mv2-debian:~#./tp.sh users.txt
Traitement de l'utilisateur toto
Changing password for user toto.
passwd: all authentication tokens updated successfully.
Traitement de l'utilisateur titi
Changing password for user titi.
passwd: all authentication tokens updated successfully.
Traitement de l'utilisateur tata
Changing password for user tata.
passwd: all authentication tokens updated successfully.
```

Exercice 23

Vous écrivez un script qui efface une liste d'utilisateurs (ainsi que leurs fichiers personnel) passée en paramètre. Résultat :

```
mv2-debian:~#./tp.sh titi hochon tata toto
traitement de l'utilisateur titi
traitement de l'utilisateur hochon
userdel : l'utilisateur hochon n'existe pas
traitement de l'utilisateur tata
traitement de l'utilisateur toto
```

Dépannage

Tout d'abord, soyez **très** rigoureux sur la syntaxe. N'oubliez pas d'espace, de parentèse, de crochet, etc.
Consultez le man de bash pour obtenir des informations.

Vous pouvez exécuter le script en mode débogage en utilisant **`bash -xv nom_du_script`**
Enfin, ce document contient des informations très détaillées sur le sujet :
http://www.freeos.com/guides/lsst/

Le langage de script bash offre beaucoup de possibilités de langages de programmation classiques, mais son utilisation est orientée « système ». On peut utiliser des variables, des saisies, des affichages, des conditions et des boucles.

Si vous voulez approfondir

J'espère vous avoir communiqué les bases du langage bash et vous en avoir montré les potentialités. L'approfondissement viendra de lui-même en fonction de vos besoins d'administrateur.
Vous savez que Unix est basé sur des scripts. Consultez par exemple, le contenu des répertoires /etc/rc.*. Vous verrez que ce sont les scripts lancés au démarrage de la machine.

Le firewalling

Durée approximative de cet atelier : 3 heures

▶ *Objectif*

Aborder la configuration d'un élément de sécurité désormais indispensable dans les entreprises : le pare-feu ou firewall. À la fin de cet atelier, vous saurez installer le pare-feu Mandrake Multiple Network Firewall, configurer les interfaces réseau, les zones et définir quelques règles élémentaires.

▶ *Conditions préalables*

Votre serveur Linux doit être opérationnel ainsi que les différents services que nous avons installés jusqu'à présent, en particulier http et ssh.

▶ *Considérations techniques*

Une troisième machine qui fera office de firewall est nécessaire.
Si vous ne pouvez pas disposer du même logiciel, lisez tout de même cet atelier.

Introduction

Un firewall est un appareil qui applique un ensemble de règles, qui filtrent ou modifient les paquets reçus ou émis en fonctions de certaines conditions. Sous linux, avec iptables, ces règles sont contenues dans des tables qui ont chacune un rôle particulier :

- *nat* : pour faire de la transformation d'adresses, indispensable pour partager un accès internet sur un réseau. Nous l'avons déjà utilisée dans le tout dernier atelier du tome 1 du cours 3988 ;
- *mangle* : pour modifier les en-têtes des paquets IP (nous ne l'utiliserons pas ici) ;
- *filter* : pour filtrer les paquets (çà, c'est très intéressant !).

Chaque table est associée à une ou plusieurs chaînes. Chaque chaîne intervient à un endroit spécifique :

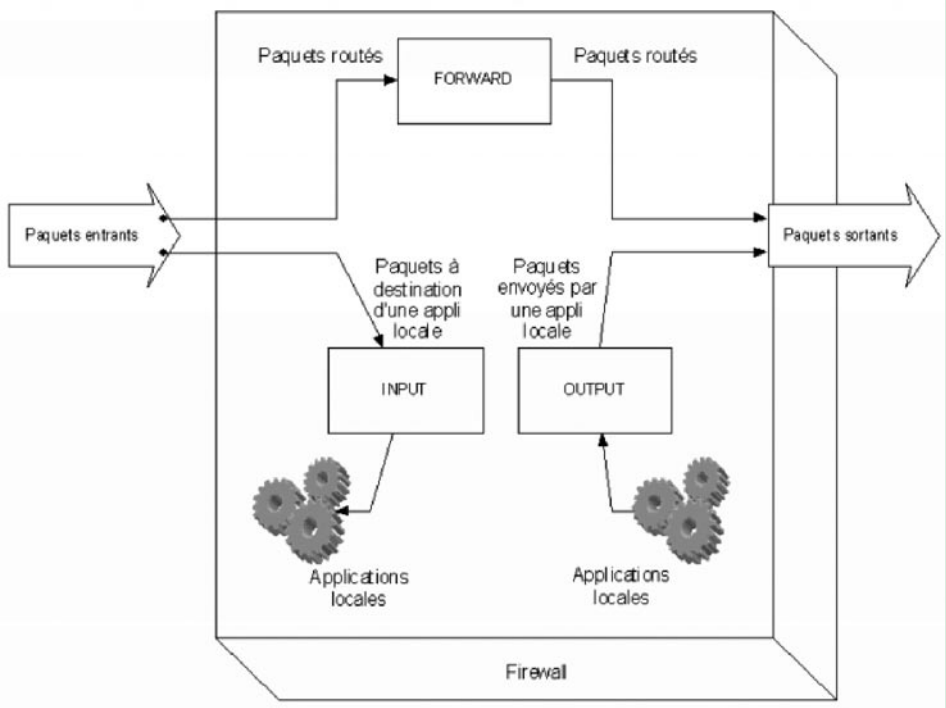


Figure 1 : architecture de netfilter

Dans cet atelier, nous nous concentrons sur la table *filter*. Les trois chaînes associées à cette table se nomment :

- INPUT : pour les paquets entrants dans le firewall à destination des applications fonctionnant dans le firewall ;
- OUPTUT : pour les paquets sortants du firewall, émis par les applications fonctionnant dans le firewall ;
- FORWARD : pour les paquets qui sont simplement transmis par le firewall, sans passer par les applications.

Ainsi, par exemple, si je veux filtrer certains paquets qui transitent par mon firewall, j'ajouterais une ou plusieurs règles à la table nommée FORWARD.

L'architecture présentée ici est celle de *Netfilter*, le firewall « classique » sous Linux car inclu dans le noyau. Comme toujours sous Linux, nous pourrions faire toutes les manipulations en ligne de commande. Mais pour diverses raisons, nous travaillerons par la suite avec une distribution Linux dédiée à la sécurité. Nous la présentons dans le paragraphe suivant.

Présentation du firewall MNF

Dans cet atelier, nous allons installer et travailler avec le firewall Mandrake Multiple Network Firewall. C'est une distribution Linux dédiée à la sécurité qui intègre nombre de fonctionnalités en sus du firewall. Vous pouvez télécharger l'image ISO sur :

ftp://ftp.lip6.fr/pub/linux/distributions/mandrake-iso/mnf/i586

MNF définit 4 zones (3 zones correspondant à des réseaux plus une zone appelée Fw et correspondant au firewall lui-même). Elle sont distinguées en fonction du degré de confiance :

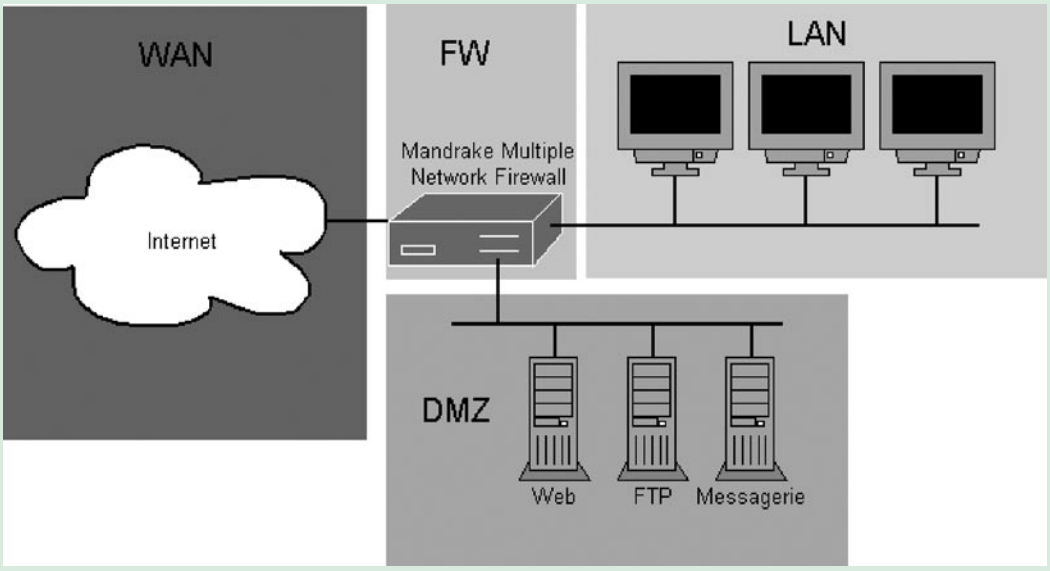


Figure 2 : les zones de MNF

Si ce document était en couleur, vous verriez :

- du rouge pour la zone WAN correspondant classiquement à Internet : la zone dont le degré de confiance est le plus faible ;
- de l'orange pour la zone DMZ. Cette zone contient généralement les machines accessibles depuis l'extérieur ;
- du vert pour le LAN car c'est la zone avec le degré de confiance maximum. Ces machines ne sont généralement pas accessibles depuis l'extérieur.

Préparation de l'atelier

• Topologie de travail

Dans cet atelier, nous allons modifier la topologie habituelle afin de se rapprocher au mieux de la réalité. Un firewall est un rempart entre différents réseaux. Vous mettrez donc en œuvre la topologie suivante :

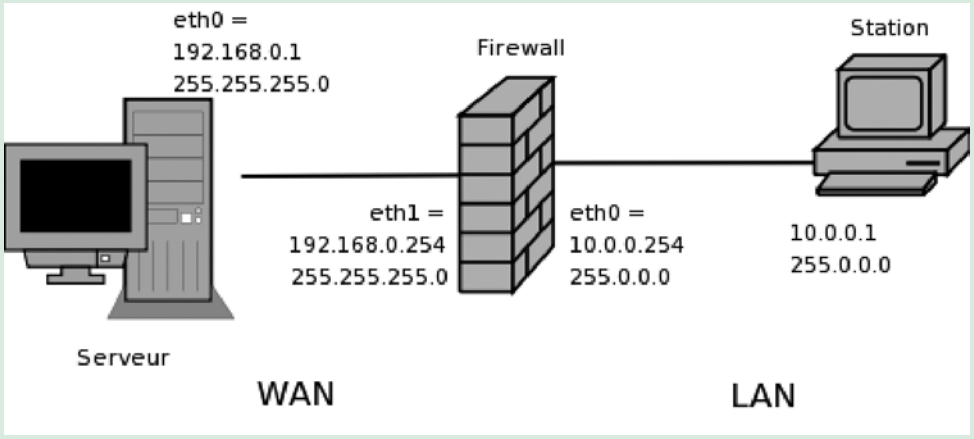


Figure 3 : topologie de travail

L'appareil qui servira de firewall aura deux cartes réseau. Chacune sera reliée à un réseau et une zone particulière. Pour être pleinement efficace, le pare-feu doit isoler physiquement les réseaux concernés.

Il faudra modifier la configuration réseau de la station et lui retirer, pour les tests, la référence au serveur DNS situé sur le serveur puisque cette machine ne fait plus partie du réseau 192.168.0.0.

Exercice 24

Quelle adresse de passerelle doit être configurée sur le serveur et sur la station ?

• Installation de MNF

Après avoir modifié les configurations réseau sur le serveur et la station, vous installez la MNF sur l'ordinateur qui fera office de pare-feu. L'installation ne doit pas poser de problèmes particuliers. Vous constaterez que l'on vous demandera deux mots de passe :

- admin : c'est l'utilisateur avec lequel vous vous connecterez pour configurer le firewall à partir de l'interface Web ;
- root : c'est comme toujours l'administrateur de la machine. **Vous ne l'utiliserez pas** pour configurer le firewall à partir de l'interface web (il ne peut de toute façon pas se connecter).

Ces deux utilisateurs ont été distingués pour des raisons de sécurité. Si le mot de passe de admin est découvert, vous gardez quand même la main sur le firewall puisque vous pourrez vous connecter en root en mode *rescue* (voir atelier sur la récupération).

Configurez bien les cartes réseau comme il est indiqué sur le schéma précédent.

À la fin de l'installation, vous apprendrez que l'administration du firewall se fera à partir de l'URL suivante :

https://10.0.0.254:8443

Elle ne pourra donc se faire qu'à partir de la station. En effet, l'interface d'administration est par défaut eth0. Vous ne pouvez donc pas configurer le pare-feu à partir de son interface eth1.

• Validation de l'installation

Nous allons faire un certain nombre de manipulations afin de vérifier que tout est correctement installé. Si ce n'est pas le cas, reportez-vous à la section « Dépannage ».

Allez sur la station, ouvrez le navigateur web et tapez dans la zone d'adresse :

https://10.0.0.254:8443

Vous devez obtenir un écran de *login* :

Figure 4 : écran de login de MNF

Vous vous connectez en **admin**. Vous arrivez alors sur un écran d'accueil qui résume les différentes fonctionnalités de l'appareil :

MandrakeSecurity
Bienvenue au centre de configuration du Mandrake Network Firewall !
Configuration système
Accès à Internet
Services
Configuration du pare-feu
Ce puissant logiciel a été conçu pour répondre aux besoins en sécurité de nos clients. Il permet de sécuriser efficacement un ou plusieurs réseaux d'entreprise.
Mandrake Network Firewall, utilise un noyau Linux de la série 2.4, et fournit des fonctionnalités avancées comme :
Filtrage adaptif (Stateful Firewalling)

Figure 5 : écran d'accueil de MNF

• Dépannage

Tout d'abord, il existe une documentation du firewall sous la forme d'un fichier pdf à télécharger sur le site de Mandrake. Il contient de précieuses informations alors utilisez-le !

Je n'arrive pas à faire des pings

Il est inutile d'essayer d'envoyer des *pings* sur le firewall. Pour des raisons de sécurité, celui-ci est configuré par défaut pour ne pas répondre. D'autre part, le logiciel possède en standard un certain nombre de règles déjà configurées. Les *pings* sont bloqués entre les zones WAN et LAN. Il est donc inutile d'essayer de pinger au travers du firewall.

Par contre, vous pouvez vous connecter en ligne de commande directement sur le firewall et faire des *pings* avec les machines de chaque côté. Il faut d'abord se loguer en admin puis en root (commande *su*).

Je n'obtiens pas l'écran de login

Au login, directement sur le firewall, connectez-vous en admin puis au $, connectez-vous en root (commande *su*). Vérifiez ensuite la configuration réseau avec la commande *ifconfig*. Toujours avec *ifconfig*, corrigez si nécessaire (il faudra ensuite, refaire la configuration des cartes dans le firewall). Vous devriez maintenant pouvoir vous connecter à partir de la station.

Vérifiez également avec la commande *netstat* qu'un serveur écoute bien sur le port TCP 8443. Si ce n'est pas le cas, redémarrez le serveur. Si cela ne fonctionne toujours pas, le plus simple est peut-être de réinstaller en reprenant pas à pas.

Le login ne fonctionne pas à cause des cookies

Et pourtant, le navigateur autorise les cookies ! Une seule cause, l'heure entre le firewall et la station diffère. Il faut que les deux horloges soient à peu près synchronisées (on n'est pas à la seconde près quand même !). Modifiez l'heure sur la station et/ou sur le firewall (en ligne de commande avec la commande *date*).

Les images ne s'affichent pas

J'ai eu ce problème parce que le navigateur était trop vieux (ou trop bogué ?). J'ai mis à jour mon vieux ie4 avec un ie5 et ça a marché. Ce problème n'est pas trop grave dans la mesure où la configuration du firewall demeure possible.

Configuration système

Ne pas utiliser les boutons de navigation (retour à la page précédente) du navigateur mais ceux proposés par MNF dans les pages html. Toujours faire APPLIQUER après une modification de configuration.

Allez dans « configuration système » puis dans « cartes réseau ». Affectez (ou vérifiez) la carte eth0 dans la zone LAN et la carte eth1 dans la zone WAN :

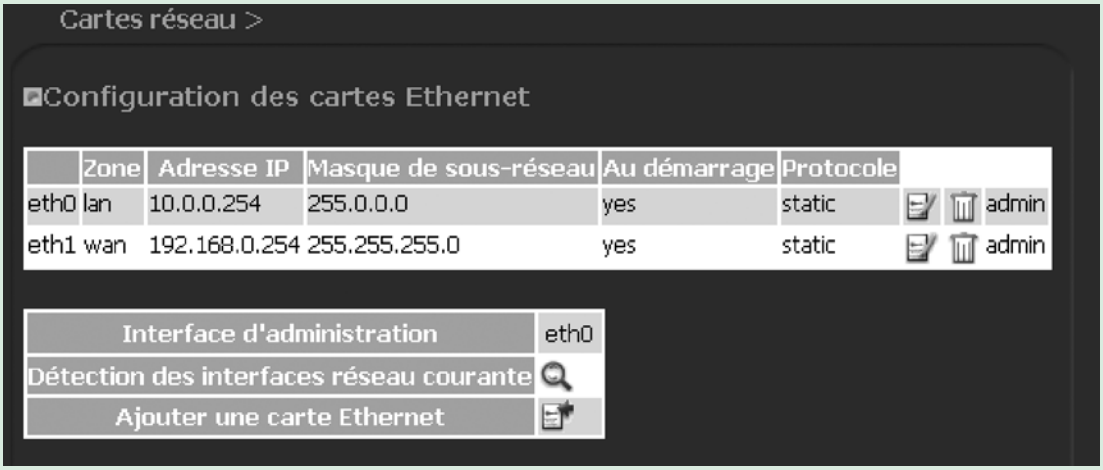


Figure 6 : configuration des cartes réseau

Contrôlez également la configuration IP des cartes. Enfin, constatez que l'interface d'administration a bien été positionnée sur eth0.

Vérifiez la configuration des zones du pare-feu dans « Configuration du pare-feu/ Définition des zones » :

Figure 7 : configuration des zones

Si tout est correctement configuré, vous devez avoir accès au serveur web à partir de la station. En effet, comme il le sera dit plus loin, le firewall possède un certain nombre de règles par défaut. En particulier, il laisse passer les flux http du WAN vers le LAN. Essayez de vous connecter à votre serveur web depuis la station :

Page de test avec mot de passe - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Page d'accueil
Cliquez ici pour vous connecter

Figure 8 : écran d'accueil affiché au travers du firewall

L'écran ci-dessus montre une page d'accueil créée dans un précédent atelier, située sur le serveur web.

Les règles

• Définition d'une règle

Une règle est composée d'un ensemble de paramètres qui constituent les conditions :

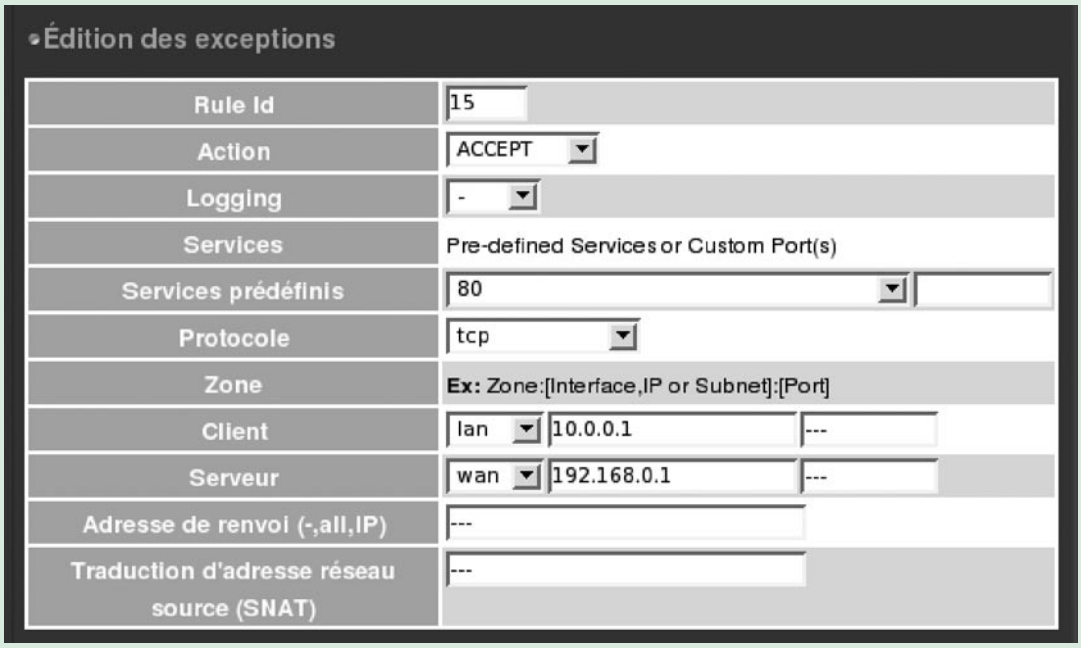


Figure 9 : contenu d'une règle de firewall

Observez sur la page ci-dessous la signification de ces différents paramètres.

Champ	Signification
ID de la règle	Le numéro identifiant de la règle.
Action	L'action entreprise pour les paquets IP qui correspondent avec cette règle (voir tableau suivant).
Logging	Si ce champ vaut « info », une trace de la connexion est enregistrée dans le journal système.
Services prédéfinis	Protocole applicatif à contrôler (http, ftp, etc.) ou son numéro de port.
Protocole	Le type de protocole à contrôler (généralement TCP, UDP, ICMP). Doit être cohérent avec le service.
Client	La zone depuis laquelle la paquet provient. La correspondance peut être réduite en spécifiant une adresse IP et/ou un masque réseau précis. Laissez « - » dans le champ pour que la requête corresponde à n'importe quel port ou adresse IP.
Serveur	La zone vers laquelle le paquet est dirigé. La correspondance peut être réduite en spécifiant une adresse IP et/ou un masque réseau précis. Laissez « - » dans le champ pour que la requête corresponde à n'importe quel port ou adresse IP.

Lorsqu'un paquet remplit une règle, il faut dire au pare-feu ce qu'il doit en faire :

Action	Signification
Accepter *(ACCEPT)*	**Le paquet est autorisé.**
Rejeter *(REJECT)*	**Le paquet est détruit et un message « destination impossible à joindre » est retourné au client.**
Retirer *(DROP)*	**Le paquet est détruit, aucune réponse n'est faite au client.**

Exercice 25

En complétant le tableau suivant, donnez la règle qui interdit l'accès au web sur le WAN depuis le LAN et la règle qui autorise SSH de l'extérieur vers le lan. Vous avez le droit de consulter la documentation du logiciel, bien sûr.

ID	Action	Logging	Service	Client	Serveur

• Traitement des règles : la politique par défaut

Tout d'abord, on définit une politique par défaut. En général, il s'agit de refuser tous les paquets. C'est la politique par défaut proposée par MNF. Allez dans « Configuration du pare-feu / Règles par défaut » :

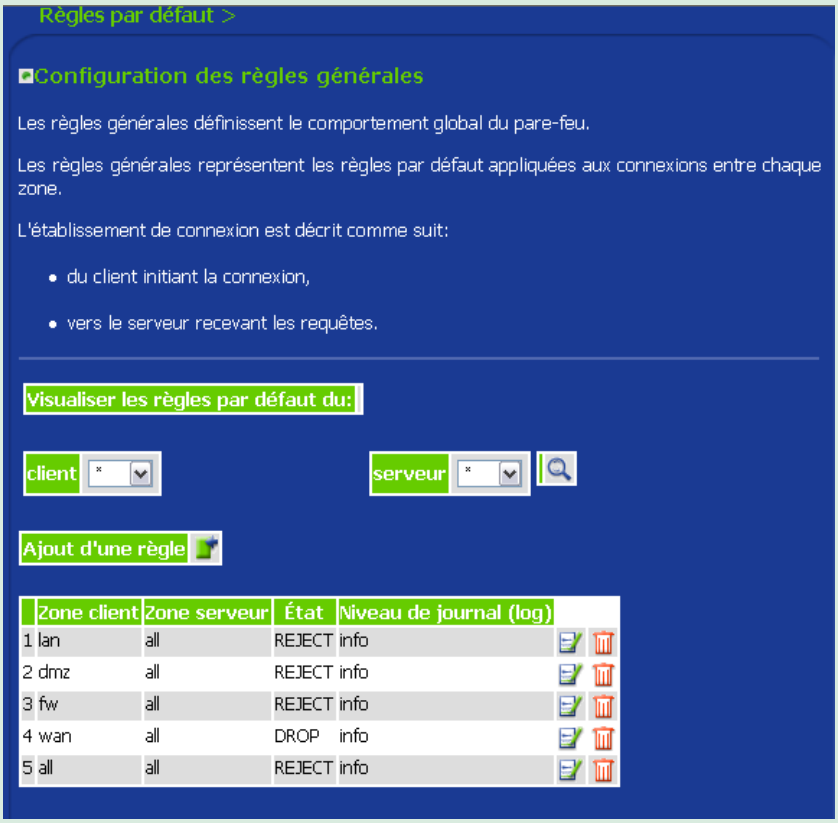

Règles par défaut >
Configuration des règles générales
Les règles générales définissent le comportement global du pare-feu.
Les règles générales représentent les règles par défaut appliquées aux connexions entre chaque zone.
L'établissement de connexion est décrit comme suit:
du client initiant la connexion,
vers le serveur recevant les requêtes.
Visualiser les règles par défaut du:
client
serveur
Ajout d'une règle
Zone client Zone serveur État Niveau de journal (log)
1 lan all REJECT info
2 dmz all REJECT info
3 fw all REJECT info
4 wan all DROP info
5 all all REJECT info

Figure 10 : règles par défaut de MNF

Toutes les règles par défaut sont à REJECT ou à DROP.

Il est important de noter que **l'ordre des règles est fondamental :** lorsqu'une règle est remplie, l'action est réalisée puis le traitement est terminé (les règles suivantes ne sont pas examinées).

• Traitement des règles : les exceptions

Partant de la politique par défaut, la configuration du firewall consiste à définir des exceptions qui vont autoriser un certain trafic. Allez dans « Configuration du pare-feu / Exceptions » :

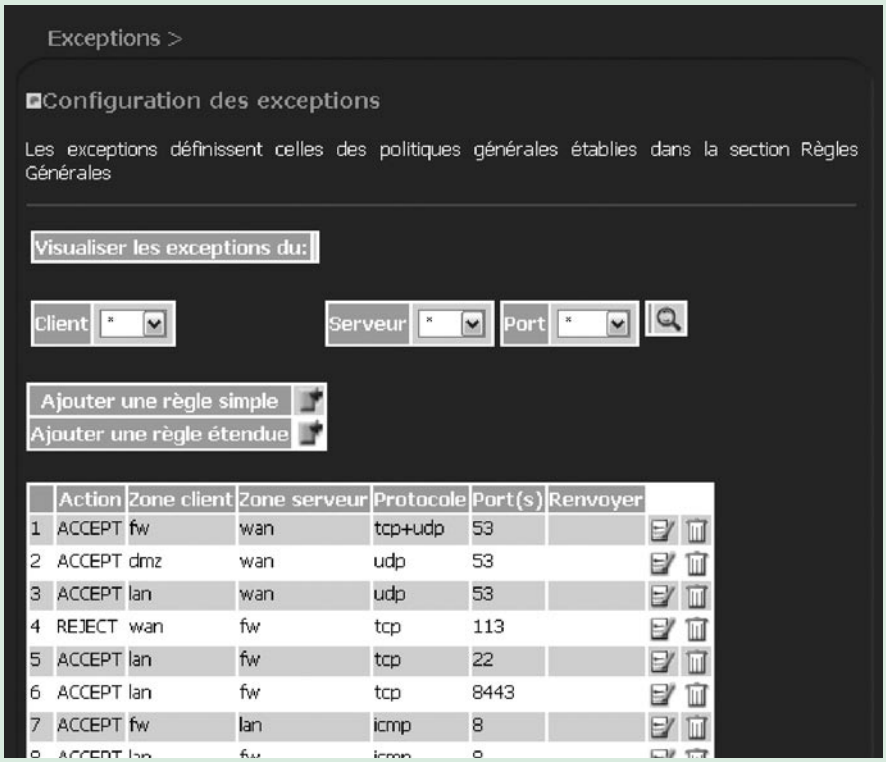

Exceptions >
Configuration des exceptions
Les exceptions définissent celles des politiques générales établies dans la section Règles Générales
Visualiser les exceptions du:
Client
Serveur
Port
Ajouter une règle simple
Ajouter une règle étendue
Action Zone client Zone serveur Protocole Port(s) Renvoyer
1 ACCEPT fw wan tcp+udp 53
2 ACCEPT dmz wan udp 53
3 ACCEPT lan wan udp 53
4 REJECT wan fw tcp 113
5 ACCEPT lan fw tcp 22
6 ACCEPT lan fw tcp 8443
7 ACCEPT fw lan icmp 8

Figure 11 : exceptions aux règles par défaut

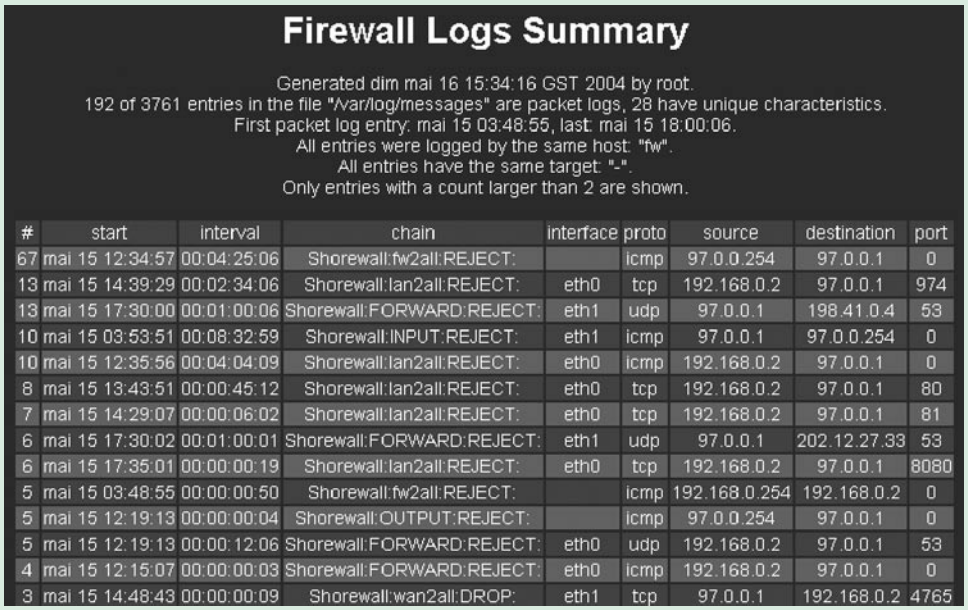
• Traitement des règles : dans quel ordre ?

Le firewall fonctionne de cette façon :

1. en premier lieu, les exceptions sont traitées dans l'ordre où elles ont été définies. Si une exception est remplie, l'action est réalisée puis le traitement est abandonné ;
2. si aucune exception n'a été remplie, on traite les politiques par défaut, dans l'ordre où elles ont été définies ;
3. si aucune politique par défaut n'a été remplie, le paquet est transmis.

• Traitement des règles : le journal

Le firewall génère une trace des connexions, ce qui est pratique pour vérifier et corriger le fonctionnement de l'appareil (menu « Surveillance/Journaux ») :



Firewall Logs Summary

Generated dim mai 16 15:34:16 GST 2004 by root.
192 of 3761 entries in the file "/var/log/messages" are packet logs, 28 have unique characteristics.
First packet log entry: mai 15 03:48:55, last: mai 15 18:00:06.
All entries were logged by the same host: "fw".
All entries have the same target: "-".
Only entries with a count larger than 2 are shown.

#	start	interval	chain	interface	proto	source	destination	port
67	mai 15 12:34:57	00:04:25:06	Shorewall:fw2all:REJECT:		icmp	97.0.0.254	97.0.0.1	0
13	mai 15 14:39:29	00:02:34:06	Shorewall:lan2all:REJECT:	eth0	tcp	192.168.0.2	97.0.0.1	974
13	mai 15 17:30:00	00:01:00:06	Shorewall:FORWARD:REJECT:	eth1	udp	97.0.0.1	198.41.0.4	53
10	mai 15 03:53:51	00:08:32:59	Shorewall:INPUT:REJECT:	eth1	icmp	97.0.0.1	97.0.0.254	0
10	mai 15 12:35:56	00:04:04:09	Shorewall:lan2all:REJECT:	eth0	icmp	192.168.0.2	97.0.0.1	0
8	mai 15 13:43:51	00:00:45:12	Shorewall:lan2all:REJECT:	eth0	tcp	192.168.0.2	97.0.0.1	80
7	mai 15 14:29:07	00:00:06:02	Shorewall:lan2all:REJECT:	eth0	tcp	192.168.0.2	97.0.0.1	81
6	mai 15 17:30:02	00:01:00:01	Shorewall:FORWARD:REJECT:	eth1	udp	97.0.0.1	202.12.27.33	53
6	mai 15 17:35:01	00:00:00:19	Shorewall:lan2all:REJECT:	eth0	tcp	192.168.0.2	97.0.0.1	8080
5	mai 15 03:48:55	00:00:00:50	Shorewall:fw2all:REJECT:		icmp	192.168.0.254	192.168.0.2	0
5	mai 15 12:19:13	00:00:00:04	Shorewall:OUTPUT:REJECT:		icmp	97.0.0.254	97.0.0.1	0
5	mai 15 12:19:13	00:00:12:06	Shorewall:FORWARD:REJECT:	eth0	udp	192.168.0.2	97.0.0.1	53
4	mai 15 12:15:07	00:00:00:03	Shorewall:FORWARD:REJECT:	eth0	icmp	192.168.0.2	97.0.0.1	0
3	mai 15 14:48:43	00:00:00:09	Shorewall:wan2all:DROP:	eth1	tcp	97.0.0.1	192.168.0.2	4765

Figure 12 : un exemple de journal généré par MNF

Manipulations

Ne pas utiliser les boutons de navigation du navigateur mais ceux proposés par MNF. Toujours faire APPLIQUER après une modification de configuration.

Exercice 26

1. Autorisez les pings (ICMP, port 8) du LAN vers l'extérieur.
Pour la suite, méfiez-vous de la mise en cache réalisée par Internet Explorer.
2. Empêchez l'accès du LAN aux sites web internet.
3. Autorisez la seule machine 192.168.0.1 du LAN à accéder au site Internet 10.0.0.1.
4. Autorisez l'accès ssh vers Internet, mais uniquement vers le serveur 192.168.0.1.

Un firewall isole physiquement plusieurs réseaux et analyse le trafic IP entre ces réseaux.
La politique par défaut est, généralement, de bloquer tout le trafic. On ajoute ensuite un certain nombre de règles considérées comme des « exceptions » à la politique par défaut.
Ces exceptions autorisent un certain trafic. Chaque paquet est analysé et comparé avec chaque règle en respectant l'ordre. Lorsqu'une correspondance est trouvée, l'action est exécutée et le traitement est terminé. Si aucune correspondance n'est trouvée, les politiques par défaut sont appliquées, le paquet est donc détruit.
Les conditions des règles portent sur les zones, les adresses IP (ou des plages), le sens des paquets et les services concernés.

Si vous voulez approfondir

Nous n'avons qu'effleuré les capacités du pare-feu mais je pense que vous savez maintenant l'essentiel. En fonction de situations auxquelles vous serez confronté, vous serez amené à aller plus loin.

Séquence 5

Filtrage, NAT et PAT

Durée indicative : 2 heures

Cette séquence explique les techniques NAT et PAT qui sont complémentaires au filtrage, et surtout comment elles se combinent entre elles.

▶ *Prérequis*

Avoir compris la notion de filtrage par un firewall.

▶ *Contenu*

1. **Introduction** 113
2. **Rappel sur le filtrage** 114
3. **La translation d'adresse : NAT** 115
4. **La translation de port : PAT** 116
5. **Combinaison NAT/PAT par l'exemple** 117
6. **Résolution de l'exercice étude de cas 2004** 118

▶ *Capacités attendues*

- **Être capable d'expliquer le principe de ces techniques de sécurité.**
- **Être capable de résoudre des exercices sur ces techniques, voire les combinant.**

1. Introduction

Le sujet d'étude de cas de la Nouvelle-Calédonie 2004 proposait un exercice sur la sécurité très intéressant, mais pas si facile.

Extrait de l'étude de cas

Pour maintenir la sécurité interne de l'entreprise VISTE, la société de service propose de ne pas mettre les serveurs HTTP et Mail sur le réseau interne. Sur le routeur Général, les règles suivantes ont été écrites :

Règles NAT *(Network Address Translation)* **PAT** *(Port Address Translation)* **appliquées sur l'interface 172.16.0.129**

IP	Port
172.16.0.66	2020

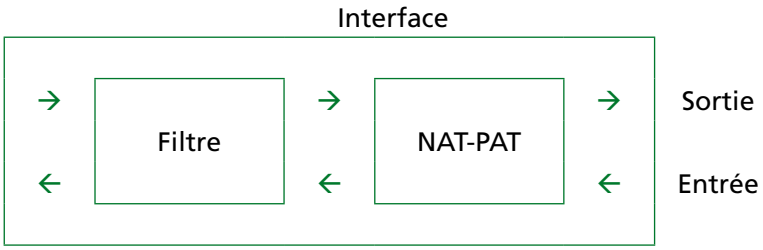
IP	Port
192.168.0.5	5600

Règles de filtrage appliquées sur l'interface 172.16.0.129

N° règle	Interface	IP Source	Port Source	IP Destination	Port Destination	Action
1	172.16.0.129	*	*	*	*	Refusé

Les règles de filtrage s'appliquent dans l'ordre de leur numéro.

Principes d'association des règles de filtrage et de NAT-PAT sur une interface.



- Sur un paquet **en sortie** d'une interface, on applique d'abord les règles de filtrage puis les règles NAT-PAT.
- Sur un paquet **en entrée** d'une interface, on applique d'abord les règles NAT-PAT puis les règles de filtrage.

Le serveur HTTP utilise le **port 1060** pour communiquer et le serveur de bases de données le **port 2020**.

Comment faire pour ne pas s'y perdre ?
Réponse : nous allons rappeler ce qu'est le filtrage, puis découvrir la translation d'adresse, puis la translation de port, enfin nous verrons que la combinaison de tout ça n'est qu'après tout, juste une question de rigueur intellectuelle (qualité très utile le jour J).

2. Rappel sur le filtrage

Nous avons vu le principe du filtrage dans la séquence 15 du cours 3988.

Une trame contient principalement : les adresses MAC et IP de la source et du destinataire, les ports source et destination des services (et en plus maintenant vous savez analyser une trame, refaites donc une petite capture pour repérer ces informations).

On peut donc filtrer sur ces différents critères, voyons un peu comment peut se représenter une table de filtrage (théorique, chaque logiciel adopte parfois un formalisme différent) :

N°	Interface	@ source	Port source	@ destination	Port destination	Action

– Les règles portent un numéro d'ordre (1, 2, 3 etc.), en effet la table de filtrage sera lue ligne par ligne, la première règle applicable sera appliquée.
– L'interface désigne le connecteur réseau sur laquelle s'applique la règle.
– Les @ source, port source, @ destination et port destination permettent de fixer les conditions de filtrage.
– Enfin, si la condition est remplie, on applique l'action (accepter : on laisse passer le message, refuser : on ne laisse pas passer).

Vous pouvez très utilement reprendre les exemples de la séquence 15 du cours 3988 pour vous rafraîchir la mémoire, c'est même conseillé…
Bon admettons que le passage de la trame soit accepté par les règles de filtrage… Le routeur laisse passer la trame à destination de @IP xxxx…
Mais si nous voulons plus de sécurité ?

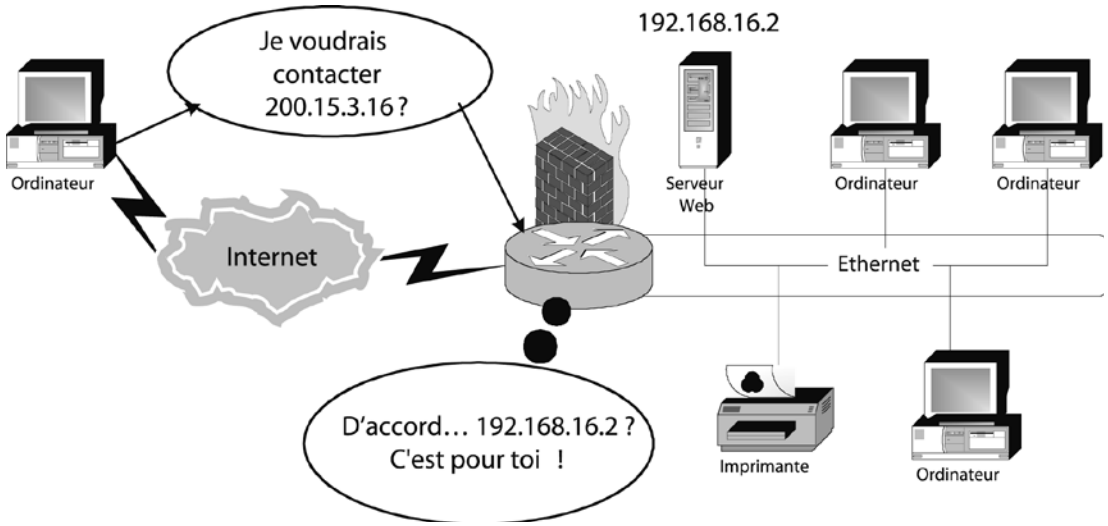
3. La translation d'adresse : NAT

NAT signifie Network Address Translation, en français : translation d'adresse réseau.

Avant de voir comment, voyons pourquoi… Dans le message accepté par le routeur quelle était l'adresse du destinataire ? La « vraie » adresse IP du destinataire, donc la source connaissait cette adresse ! Est-ce toujours souhaitable ?

Si la source est l'administrateur d'un serveur web, par exemple, et qu'il veut l'administrer de chez lui, aucun souci. Mais si c'est l'internaute *lambda*, avez-vous envie qu'il connaisse l'adresse IP de votre tout nouveau serveur web ? Quelles sont ses intentions ???

Comment peut-on lui cacher ? Simple : on en met une autre à la place, seul le routeur connaît la vraie adresse destination !



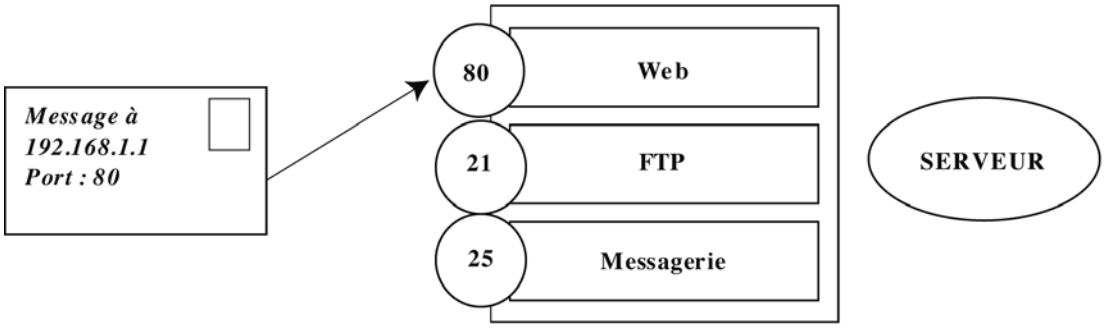
Alors comment ça marche ? Oh en théorie c'est très simple : le routeur désencapsule le message jusqu'au datagramme IP = il « enlève » toutes les infos des couches 1&2… là il remplace l'@IP du destinataire par la « vraie » qu'il a dans sa table, CQFD.

Et au retour lorsque le serveur web répond ? Eh bien il fait la même chose… dans l'autre sens. Il remplace la « vraie IP » par une factice : le client n'y voit que du feu ! Vous allez voir tout à la fin la solution de l'étude de cas de 2004, vous allez vous dire : « finalement c'est un jeu d'enfant ! ».

4. La translation de port : PAT

Vous vous souvenez aussi de la séquence 9 ? Bien sûr…

Eh bien chaque application utilise un port de communication différent qui est identifié par un numéro de port :

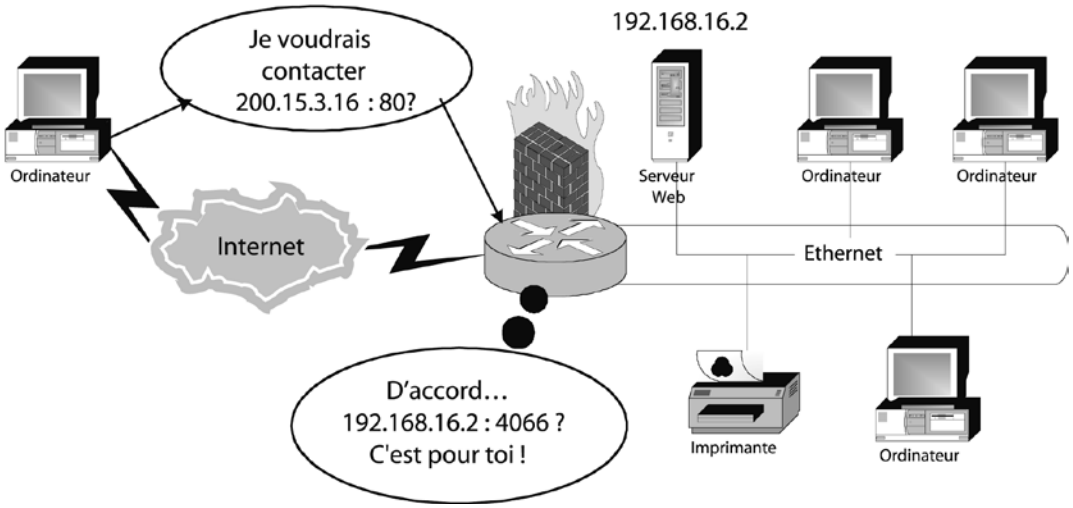


Dans l'analyse d'une trame on retrouve bien sûr le port de communication du service…

Et si on le cachait aussi le « vrai » port ?
Qui c'est qui va embêter mon serveur web si je le mets sur le port 4066 ? (évitez xx80 d'accord ?) Avant de le trouver sur 4066 il faudra que le « client » soit au courant ou… passe en revue mes ports ouverts : on parle alors de scan de ports (un des logiciels les plus connus est nmap).
Eh bien même principe que le nat : c'est le routeur qui va échanger dans la transmission externe le « vrai » port… vu de l'extérieur ce sera peut-être le 80 (port natif d'un service web), mais le vrai sera 4066 et hop !

5. Combinaison NAT/PAT par l'exemple

Prenons le schéma suivant :



Vous connaissez la notation : 200.15.3.16 :80 ? C'est surtout utilisé pour l'administration des équipements d'interconnexion : switchs, routeurs… Mais cela veut tout simplement dire : je communique avec l'hôte d'@IP xxxx sur le port yy, donc on tape xxxx :yy

Donc sur le schéma exemple vous avez remarqué : le client demande 200.15.3.16 :80 et finalement le « vrai » destinataire c'est 192.168.16.2 :4066 !!! C'est le routeur qui fait tout ça (si le fabricant a bien voulu implémenter ces fonctions bien sûr. Attention avant d'acheter même un routeur « familial » !).

En étude de cas, exemple 2004, cela peut finalement s'écrire :

Règles NAT *(Network Address Translation)* **PAT** *(Port Address Translation)* **appliquées sur l'interface 172.16.0.129**

IP	Port
172.16.0.66	2020

↔

IP	Port
192.168.0.5	5600

6. Résolution de l'exercice étude de cas 2004

Le sujet nous simplifie (ou nous rappelle gentiment les notions) :

Règles NAT *(Network Address Translation)* **PAT** *(Port Address Translation)* **appliquées sur l'interface 172.16.0.129**

IP	Port
172.16.0.66	2020

←→

IP	Port
192.168.0.5	5600

Ah mais il faut filtrer aussi ! D'accord le seul problème c'est **dans quel ordre on fait ces opérations !**

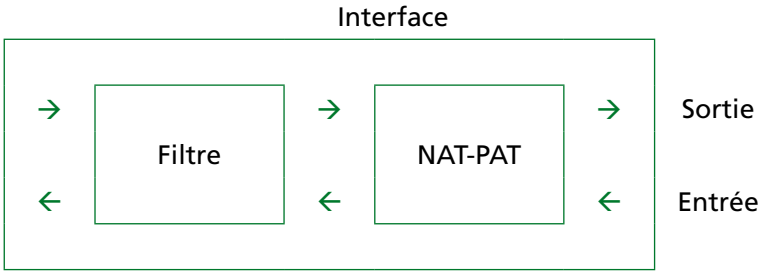
Voyons le sujet

Règles de filtrage appliquées sur l'interface 172.16.0.129

N° règle	Interface	IP Source	Port Source	IP Destination	Port Destination	**Action**
1	172.16.0.129	*	*	*	*	Refusé

Les règles de filtrage s'appliquent dans l'ordre de leur numéro.

Principes d'association des règles de filtrage et de NAT-PAT sur une interface.

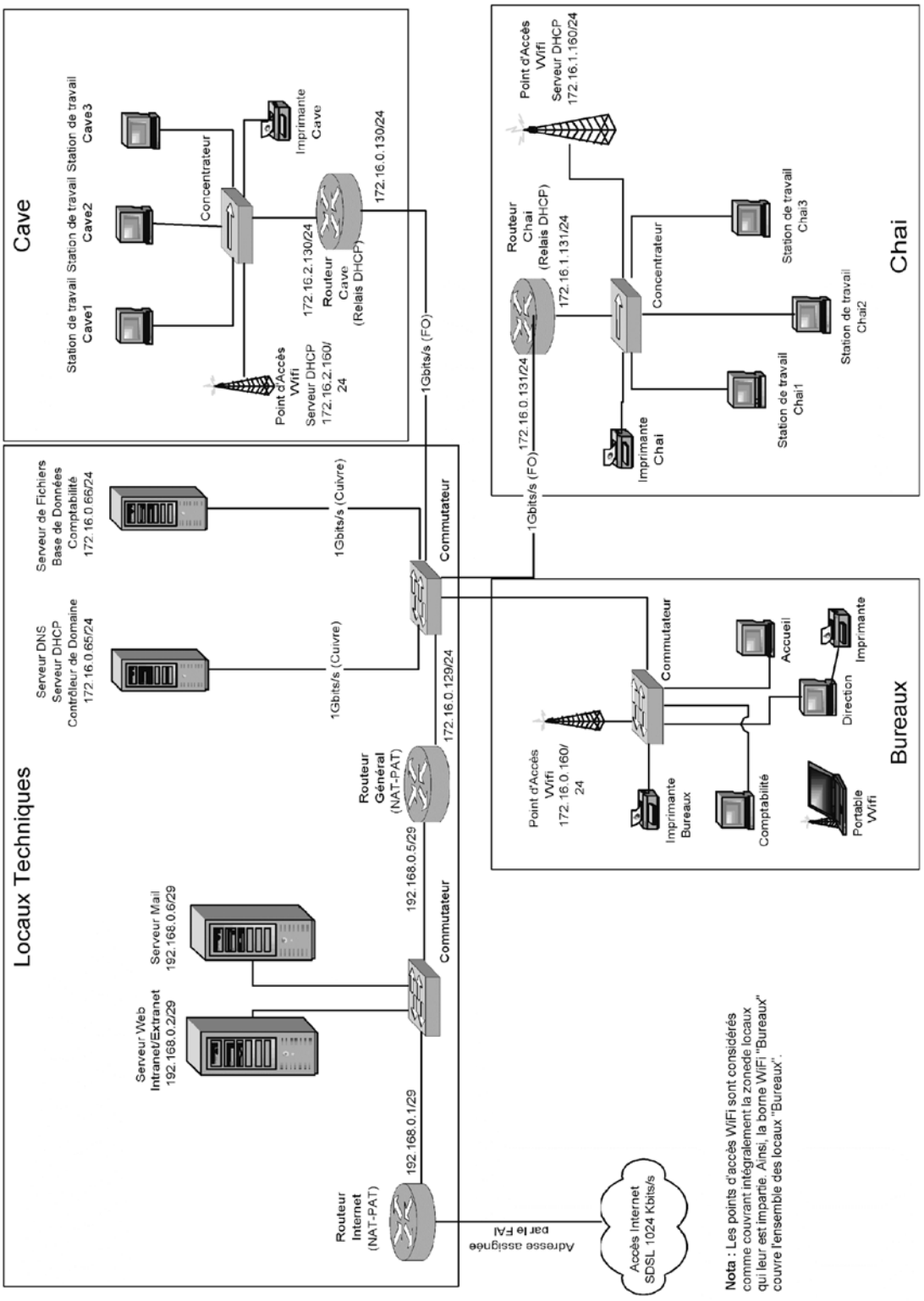


- Sur un paquet **en sortie** d'une interface, on applique d'abord les règles de filtrage puis les règles NAT-PAT.
- Sur un paquet **en entrée** d'une interface, on applique d'abord les règles NAT-PAT puis les règles de filtrage.

Le serveur HTTP utilise le **port 1060** pour communiquer et le serveur de bases de données le **port 2020**.

Question

Donner les adresses IP et les ports source et destination d'un paquet envoyé par le serveur HTTP au serveur de bases de données.



Voici le schéma (conseil : toujours avoir le schéma annexe sous les yeux = on « dégrafe » les annexes importantes !)…

Bon analysons la proposition de corrigé.

Le paquet est envoyé du serveur HTTP (192.168.0.2) sur le port 1060 comme l'indique le texte, et à destination du serveur de bases de données (172.16.0.66) pour le port 2020. Mais le paquet est pris en charge par le routeur NAT qui va assurer la translation de certaines valeurs. En effet l'adresse 172.16.0.66 et le port 2020 vont être convertis en l'adresse 192.168.0.5 et le port 5600.

Les valeurs définitives seront donc :

- Adresse IP source : 192.168.0.2
- Port source : 1060
- Adresse IP Destination : 192.168.0.5
- Port Destination : 5600

Donc d'après le schéma il y a d'abord filtrage puis application des translations… Il est juste dommage que les informations sortie et entrée soient inversées : cela peut perturber, je le reconnais…

Règle :

- on appelle entrée : les flux qui entrent sur le LAN ;
- on appelle sortie : les flux qui sortent vers l'extérieur.

Là le message du serveur http vers le serveur Base de données c'est donc un flux entrant ! Ne pas se faire piéger par le schéma…

CQFD
Mais attention il faut adopter les règles et conventions du sujet !

Proxy

Durée approximative de cet atelier : 2 heures

▶ *Objectif*

Aborder la configuration élémentaire d'un proxy afin d'optimiser, de contrôler et de filtrer l'accès à Internet.

▶ *Conditions préalables*

Idem que précédemment, cet atelier étant une suite. Nous continuons donc à travailler avec une distribution MNF correctement installée.
Si vous ne pouvez pas disposer du même logiciel, lisez tout de même cet atelier.

Introduction

• Généralités

Le proxy (Squid) de MNF propose deux fonctionnalités principales :

- **optimisation** de la bande passante sur le lien Internet, lorsque de nombreux clients sont connectés et qu'ils visitent les mêmes sites ;
- **contrôle et filtrage** de l'accès à Internet, en se basant sur les URI demandées, les horaires, les adresses IP des stations et les noms d'utilisateurs.

MNF supporte les protocoles FTP, HTTP et HTTPS (il ne gère donc pas SMTP, POP ou NNTP).

Il écoute sur un port spécifique (3128 par défaut, mais il est possible d'utiliser 8080, plus habituel pour un proxy).

•Fonctionnement

Observons, tout d'abord, l'accès à un site web sans passer par un proxy :

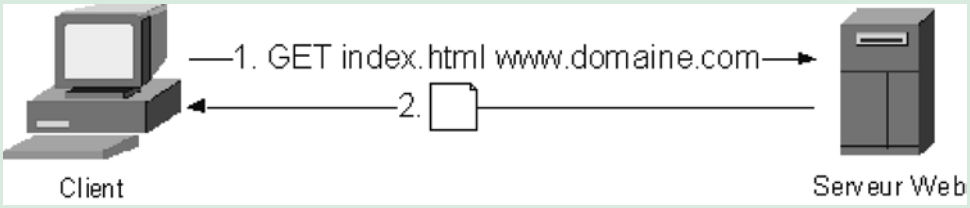


Figure 1 : Accès web sans proxy

Le client demande directement au serveur web de lui envoyer une page. Si un autre client du réseau demande la même page, le serveur web est contacté à nouveau.

Avec un proxy (littéralement, *proxy* signifie mandataire en français), toutes les requêtes passent par un intermédiaire (voir figure ci-dessous).

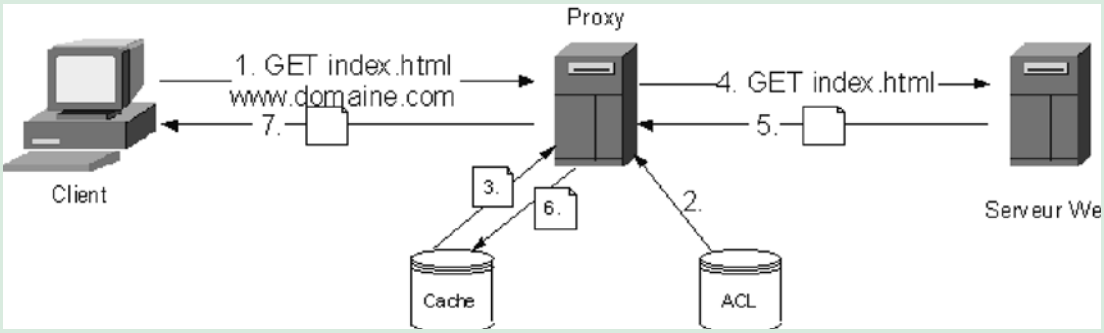


Figure 2 : Accés web avec proxy

1. Le client demande une page.
2. La requête est interceptée par le proxy. Celui-ci vérifie ses ACL*. Si la requête est interdite, un message d'erreur (non représenté sur la figure) est envoyé au client.
3. Si la requête est autorisée, le proxy vérifie si la page est déjà dans son cache (autrement dit, si elle a été demandée récemment). Si c'est le cas, elle est envoyée directement au client sans passer par le serveur web.
4. Si la page ne figure pas dans le cache, le proxy transmet la requête au serveur désiré.
5. Le serveur retourne la page.
6. Le proxy intègre cette page dans son cache.
7. Le client reçoit la page demandée.

Note : étant donné que tout le trafic IP passe par le proxy, celui-ci collecte des informations sur les utilisateurs, ce qui peut poser un problème de confidentialité.

** Access Control Lists ou Listes de Contrôle d'Accès*

• Considérations matérielles

Un proxy consomme beaucoup de ressources mémoire et CPU. Il faut donc prévoir une machine puissante, dotée d'une mémoire RAM importante et de gros disques SCSI.
De plus, il ne faut pas oublier qu'un proxy, de par son rôle d'intermédiaire, est un composant essentiel de la connectivité du réseau local à Internet. Il faut donc considérer une machine fiable équipée de composants de qualité et redondants (RAID en particulier).
Enfin, le proxy peut conserver dans son cache des informations confidentielles. Il faut donc que l'ordinateur soit sécurisé.

Activation du proxy web

Deux écrans servent à définir les paramètres généraux du proxy.

• Modes de fonctionnement

Le proxy peut être transparent ou manuel :

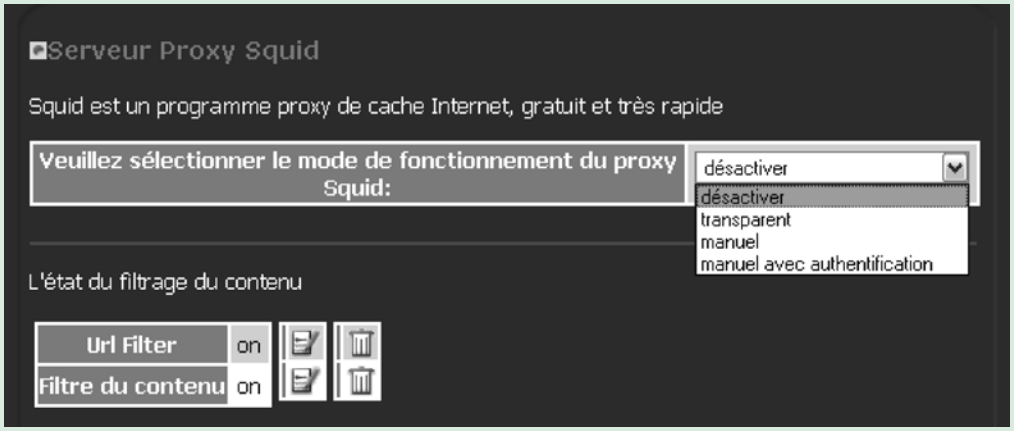


Figure 3 : configuration du type de fonctionnement

Signification des différents modes de fonctionnement :

Manuel	Une configuration est nécessaire sur les postes clients.
Manuel avec authentification	Un mot de passe est demandé à l'utilisateur pour passer au travers du proxy.
Transparent	Les utilisateurs sont obligés de passer par le proxy, aucune configuration n'est nécessaire sur les clients.

Le proxy transparent ne peut être mis en œuvre que lorsque la topologie du réseau a été conçue en conséquence :

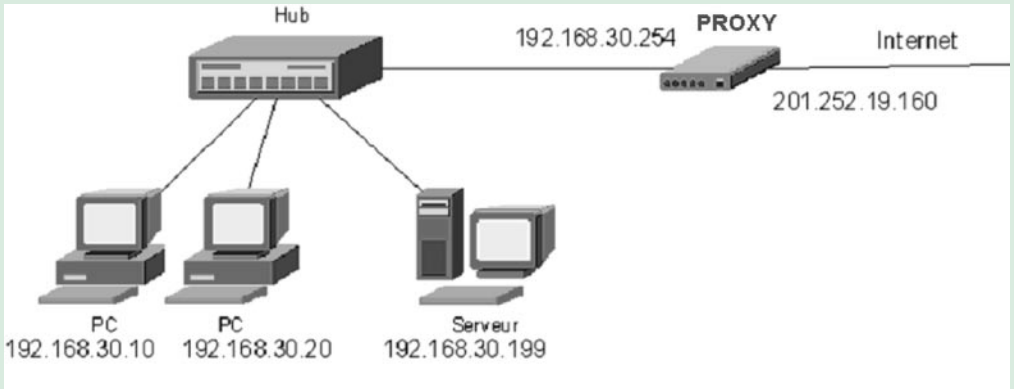


Figure 4 : le proxy transparent exige d'isoler les segments de réseaux

en proxy transparent, seuls les services web (port 80) sont traités.

• Paramètres

La configuration est relativement simple :

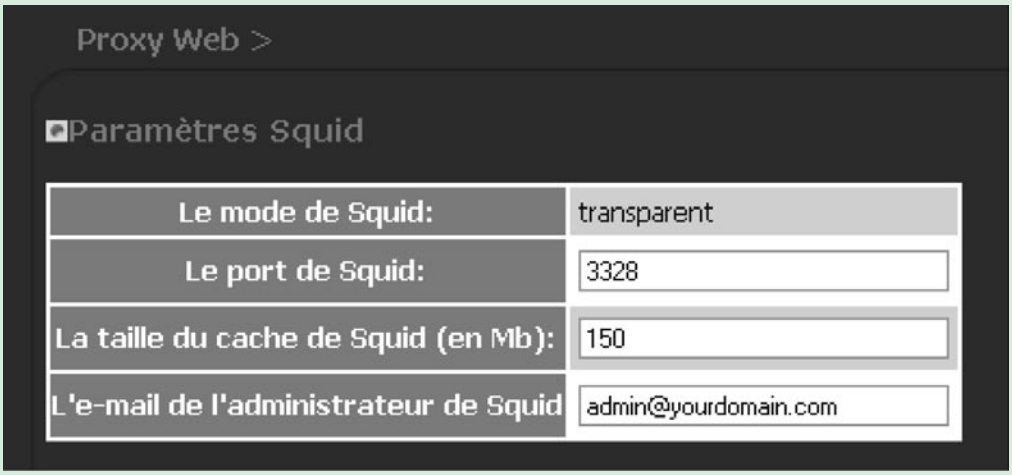


Figure 5 : paramètres du proxy

Je pense que cet écran se passe de commentaire.

Filtrage

Différents modes de filtrage sont proposés.

• Filtrage sur les adresses demandées

Les fonctionnalités suivantes sont proposées :

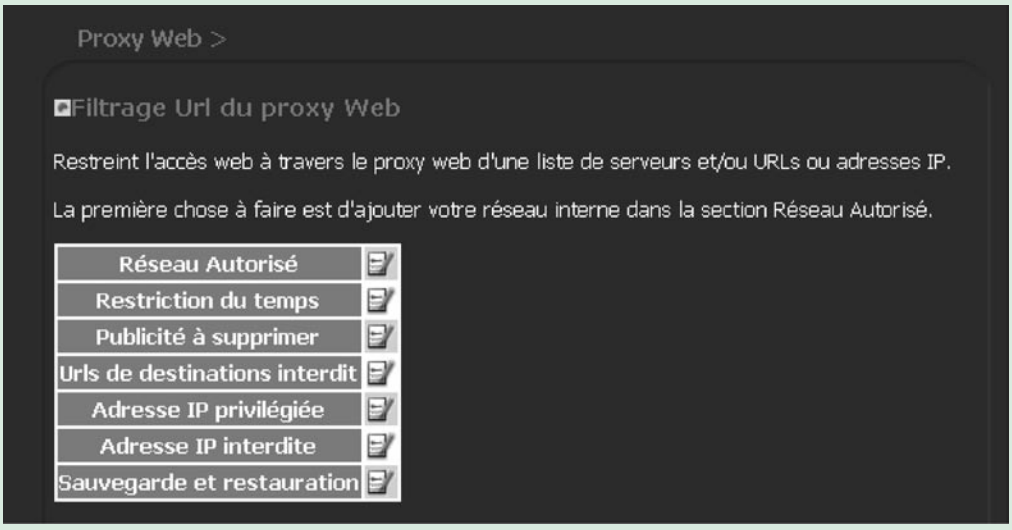

Proxy Web >
Filtrage Url du proxy Web
Restreint l'accès web à travers le proxy web d'une liste de serveurs et/ou URLs ou adresses IP.
La première chose à faire est d'ajouter votre réseau interne dans la section Réseau Autorisé.
Réseau Autorisé
Restriction du temps
Publicité à supprimer
Urls de destinations interdit
Adresse IP privilégiée
Adresse IP interdite
Sauvegarde et restauration

Figure 6 : filtrage sur les URL

Un premier niveau de filtrage peut être opéré sur les adresses IP ou DNS que les clients souhaitent atteindre :

- réseau autorisé : IP des machines autorisées à utiliser le proxy ;
- publicité à supprimer : adresses ou noms de domaine des sites publicitaires afin de supprimer les images et autres bannières sur les sites consultés ;
- URL de destinations interdit : adresses ou domaines interdits aux clients ;
- adresses IP privilégiées : IP des machines de votre réseau local qui ne seront pas soumises aux règles du proxy ;
- adresses IP interdites : IP des machines de votre réseau local qui ne sont pas autorisées à utiliser le proxy.

Une option un peu à part concerne la restriction du temps. On définit sur quelles plages horaires le proxy est actif, donc la connexion à internet.

• Filtrage sur les contenus

Les fonctionnalités suivantes sont proposées :

Proxy Web >
Paramètres du filtrage de Contenu
Configuration du Filtrage de Contenu
Liste des Mots Interdits
Liste des Extensions de Fichiers Interdites
Liste de type Mime interdits
Urls de destinations interdit
Liste des adresses IP exception
Liste des Sites exception
Liste des Utilisateurs exception
Mettre à jour les règles iptables
Redémarrer DansGuardian

Figure 7 : filtrage sur les contenus

On peut donc filtrer l'accès à des sites sur :

- mots interdits contenus dans les pages ;
- extensions de fichiers interdites (par exemple.exe,.avi,.zip,.pif...) ;
- types mime interdits (c'est une autre façon d'interdire certains types de fichiers, en particulier dans les pièces jointes aux courriers électroniques).

On peut, enfin, tout bloquer puis ne définir que quelques exceptions sur les URL, adresses IP, sites ou utilisateurs.

Manipulations

1. La première étape consiste à indiquer à MNF quelle interface correspond à internet. Nous pourrons ensuite configurer le proxy. Allez dans l'onglet « Accès à internet » puis cliquez sur « Câble et LAN ». Sélectionnez eth1 comme interface « internet ». Contrôlez la configuration proposée (en principe, il n'y a rien à changer). Cliquez sur « suivant » puis sur « appliquer ».
2. Activez le service « proxy web » en mode transparent. Allez maintenant dans « URLFilter » (services/proxy web). Ajoutez votre LAN (sous la forme 10.0.0.0/8) en tant que « réseau autorisé ».
3. Toujours dans URLFilter, interdisez l'accès à l'URL de destination 192.168.0.1. Sous Windows, essayez de vous connecter à 192.168.0.1 avec le navigateur. Le proxy vous répond impitoyablement :

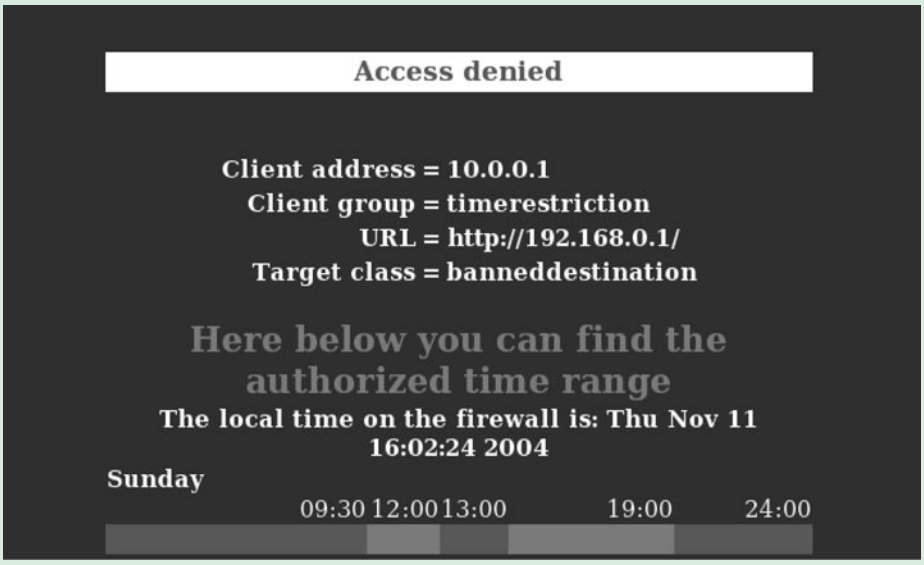

Access denied
Client address = 10.0.0.1
Client group = timerestriction
URL = http://192.168.0.1/
Target class = banneddestination
Here below you can find the authorized time range
The local time on the firewall is: Thu Nov 11 16:02:24 2004
Sunday
09:30 12:00 13:00 19:00 24:00

Figure 8 : accés à une IP bloquée

Le mot-clé important sur cette page est « *banneddestination* » qui démontre que l'URL saisie (192.168.0.1) est bannie. Les informations sur les horaires de connexion sont sans intérêt ici.

Le même exercice serait intéressant avec les noms DNS (par exemple, interdire l'accès à www.pirate.com), mais la station windows n'est plus configurée pour utiliser un serveur DNS. À faire en approfondissement !

Avant de poursuivre, supprimez le blocage de l'URL 192.168.0.1 que nous venons d'ajouter.

4. Créez à la racine du serveur web une page banni.html contenant un mot que vous souhaitez voir bloqué par le proxy (par exemple, le mot piratage) :

```
<html>
<head>
</head>
<body>
<h1>
Cette page a un contenu interdit : piratage
</h1>
</body>
</html>
```

Vérifiez que cette page est accessible avec http://192.168.0.1/banni.html depuis la station Windows. Dans « Filtre du contenu » du « Proxy web », dans la liste des « Mots interdits », ajoutez le mot que vous souhaitez voir bloqué. Tentez à nouveau d'accéder à la page :

Access denied
Access to the page :
http://192.168.0.1/banni.html/
... has been denied for the following reason :
Banned phrase found.
You are seeing this error because the page you attempted
to access contains, or is labled as containing, material that
has been deemed inappropriate.

Figure 9 : accés à une page contenant un mot interdit

5. Le blocage sur les types de fichiers.

Exercice 27

Recherchez comment bloquer l'accès à certains types de fichiers. Observez quels types de fichiers sont bloqués par défaut par le proxy.

6. Surveillance
Consultez les différents journaux produits par MNF. Visionnez les graphiques montrant l'activité du système et du réseau dans la « surveillance ». Par exemple :

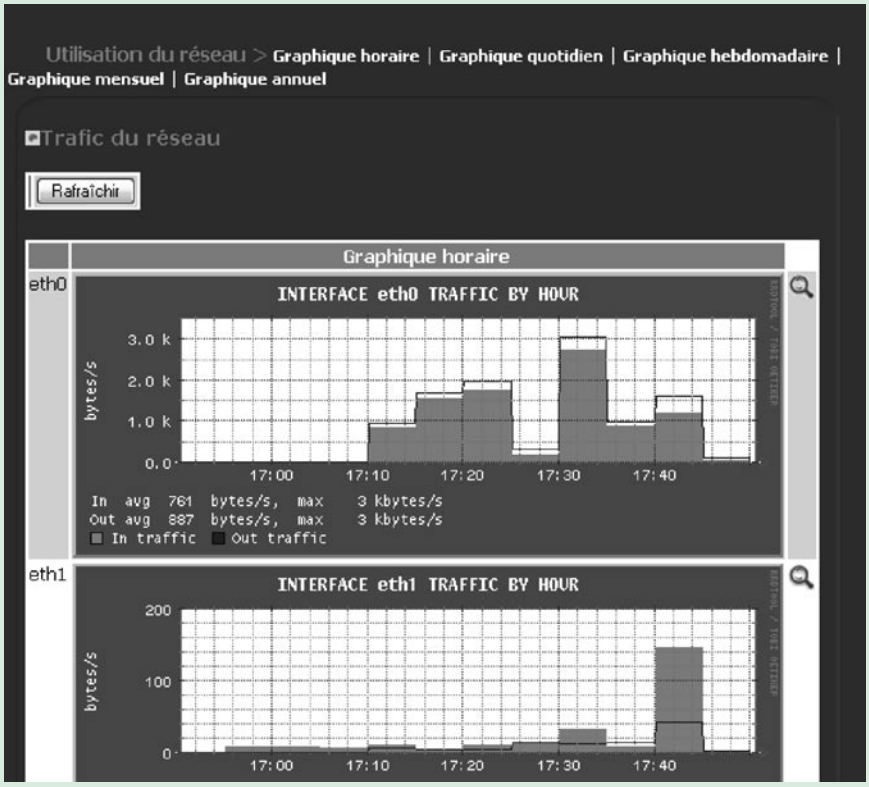


Figure 10 : les graphiques de surveillance générés par MNF

Résumé

Un proxy a pour objectif de partager un accès à Internet tout en optimisant, contrôlant et filtrant.
Un proxy peut être transparent ou non. Il optimise en gérant un cache des sites internet consultés. Il contrôle en identifiant les utilisateurs. Il filtre suivant les adresses IP, les noms de domaines DNS, les URL ainsi que sur des listes de mots-clés.

Si vous voulez approfondir

MNF dispose de nombreuses fonctionnalités supplémentaires :

– Serveur DHCP ;
– Cache DNS ;
– IDS (Intrusion Detection System) ;
– VPN (Virtual Private Network).

Explorez toutes ces possibilités !
En ce qui concerne le filtrage, il est assez complexe de définir les listes de sites « interdits ». Le mieux est de repartir des listes existantes :
http://cri.univ-tlse1.fr/documentations/cache/squidguard.html

Note : les outils de filtrage de contenu de MNF sont les logiciels libres Squidguard et Dansguardian.

Le routage IP statique

Durée approximative de cet atelier : 5 heures

▶ *Objectif*

Comprendre les principes du routage IP et l'architecture du routeur logiciel Quagga/Zebra.
Prendre en main Quagga et Zebra et mettre en pratique le routage statique.
Une fois l'atelier fini, vous saurez :

– utiliser l'interface de configuration de Quagga et Zebra ;

– saisir des routes statiques ;

– déboguer un routage incorrect.

▶ *Conditions préalables*

Cet atelier, pour être vraiment intéressant, nécessite un nombre important d'ordinateurs. Avec Virtual PC vous pourrez créer autant de machines virtuelles que vous voudrez pour tester vos manipulations.

▶ *Considérations techniques*

Ah, le routage IP ! Concept essentiel pour qui veux comprendre le fonctionnement d'internet. Essentiel aussi pour qui veut réussir l'étude de cas au BTS.
L'intérêt fondamental du logiciel Zebra/Quagga est que son langage de commande s'inspire très fortement des produits CISCO, un acteur important dans le milieu du routage. Ce sont des appareils que vous risquez de rencontrer en entreprise.

▶ *Avertissement*

Ce support de cours et le suivant s'inspirent directement d'articles que votre humble serviteur a écrit pour Linux Magazine (http://www.linuxmag-france.org). Achetez cet excellent magazine !
Et du même document mis à jour en 2007 par Philippe Latu contributeur de nombreux documents libres concernant les logiciels dans le cadre du projet inetdoc :
http://www.linux-france.org/prj/inetdoc/telechargement
Le projet inetdoc recèle de nombreux documents (articles, cours, tp…) qui complètent de façon indispensable nos ateliers !

Introduction

Dans cette introduction, nous présentons les concepts importants liés au routage. Ces points ont déjà été abordés dans le cours 3988 tome 2.

• Quelques définitions

Imaginez que vous souhaitiez rendre visite à quelqu'un. Vous connaissez son adresse et, dans votre tête, vous avez la plupart des éléments importants pour vous diriger (sortir de la maison, prendre à droite, au feu tourner à gauche, marcher jusqu'au rond-point, tourner à droite, etc.).

Dans l'univers informatique, c'est sensiblement différent. Lorsqu'un ordinateur émet un paquet de données à destination d'un ordinateur situé dans un autre réseau, **il ne sait pas quelle route il va prendre.** La seule chose connue est l'adresse de destination du paquet ainsi que l'adresse d'une machine à proximité, située dans son réseau, et qui joue le rôle de la porte. On l'appelle la passerelle. Elle oriente le paquet vers le prochain carrefour. À ce carrefour, il y a un appareil qui oriente le paquet vers le prochain carrefour. À ce nouveau carrefour, il y a un nouvel appareil qui oriente le paquet vers le prochain carrefour. Et ainsi de suite jusqu'à arriver à la passerelle du réseau de destination qui remet le paquet dans le réseau.

Le routeur

L'appareil dont nous parlons s'appelle un **routeur** (la passerelle est également un routeur). Le routage consiste à faire circuler de routeur en routeur les paquets de données.

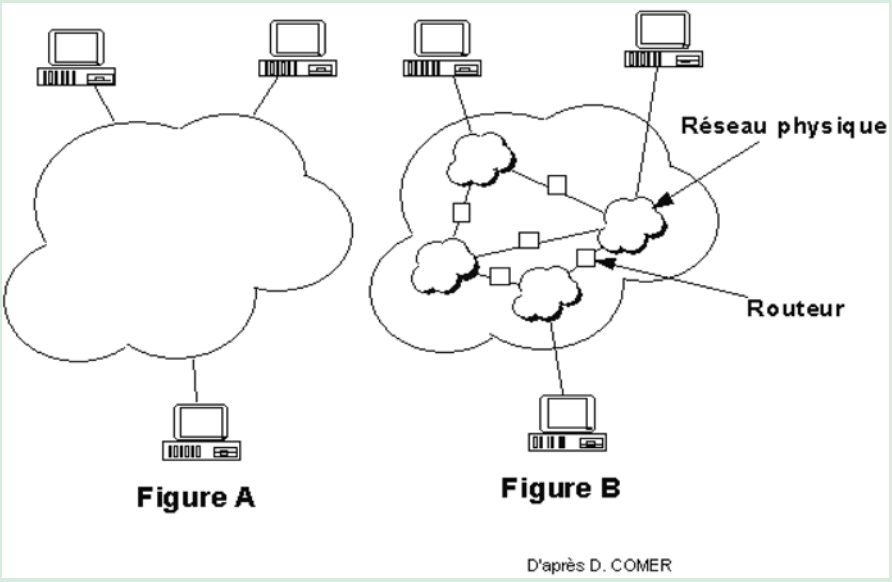


Figure 1 : architecture d'internet

Eh oui, la magie d'internet fait ressembler le réseau à la figure A. Nous avons l'impression d'être tous directement reliés. La réalité est plutôt comme sur la figure B. Un maillage de routeurs gérés par divers opérateurs.

Pour donner une définition plus technique du routeur, disons que tout appareil qui possède au moins deux interfaces réseau et qui est capable d'émettre sur une interface, un paquet reçu sur une autre interface, en fonction de règles définies dans une table de routage est un routeur. Une machine sous Linux avec deux cartes réseau peut être un routeur !

• La table de routage

L'administration d'un routeur consiste à configurer les routes d'un routeur. Une **route** est définie par un réseau de destination et l'adresse d'un routeur voisin, prochaine étape vers le réseau de destination. Une **table de routage** est une liste de routes utilisée par le routeur pour prendre des décisions quant à la direction à donner pour un paquet reçu sur l'une de ses interfaces.

Par exemple, une table de routage peut ressembler à cela :

Réseau	Masque	Passerelle	Interface
Localhost	255.0.0.0	127.0.0.1	lo
192.168.0.0	255.255.255.0	192.168.0.254	eth0
192.168.1.0	255.255.255.0	192.168.0.254	eth1

Le **routage statique** par opposition au routage dynamique, consiste à saisir manuellement les routes dans le routeur (c'est très formateur pour comprendre le fonctionnement !).

• Prise de décision

Il faut tout d'abord se rappeler que tous les paquets IP émis par une machine ont la tête suivante :

Figure 2 : structure d'un paquet

L'en-tête contient (entre autres), les adresses IP de l'émetteur et du destinataire.
Lorsqu'un routeur reçoit un paquet sur l'une de ses interfaces. Il extrait l'adresse IP de destination. Puis, il parcourt la table de routage à la recherche d'une correspondance en appliquant le masque réseau de chaque ligne de la table de routage à l'adresse IP de destination.
Si une correspondance est trouvée, il émet le paquet sur l'interface réseau concernée. Sinon, deux cas peuvent se présenter. Soit il connaît un routeur par défaut et il lui transmet le paquet, soit il le détruit (sans en informer l'émetteur).

Prenons tout d'abord un cas simple.

Exercice 28

Soit la table de routage suivante :

Réseau	Masque	Passerelle	Interface
localhost	255.0.0.0	127.0.0.1	lo
172.16.0.0	255.255.0.0	172.16.0.254	eth0
192.168.0.0	255.255.255.0	192.168.0.254	eth1
192.168.1.0	255.255.255.0	192.168.0.254	eth2

Pour une adresse de destination 192.168.0.35, indiquez et **démontrez par le calcul** sur quelle interface du routeur, le paquet sera transmis.

Exercice 29

Continuons avec un cas plus délicat. Du fait que les masque réseaux n'ont pas forcément une longueur fixe, on peut très bien avoir une table de routage de ce type :

Réseau	Masque	Passerelle	Interface
11.1.2.0	255.255.255.0	11.1.2.254	eth0
11.1.0.0	255.255.0.0	11.1.0.254	eth1
11.0.0.0	255.0.0.0	11.0.0.254	eth2
192.168.1.0	255.255.255.0	192.168.0.254	eth2

Supposons qu'un paquet ait une adresse de destination égale à 11.1.2.5. Selon vous, sur quelle interface sera-t-il transmis ?

Puisque vous avez compris la théorie et que vous savez comment fonctionne un routeur, passons maintenant aux manipulations.

Contexte de l'atelier

• Topologie de travail

Dans l'idéal, il faudrait que vous soyez en mesure de monter la topologie ci-dessous :

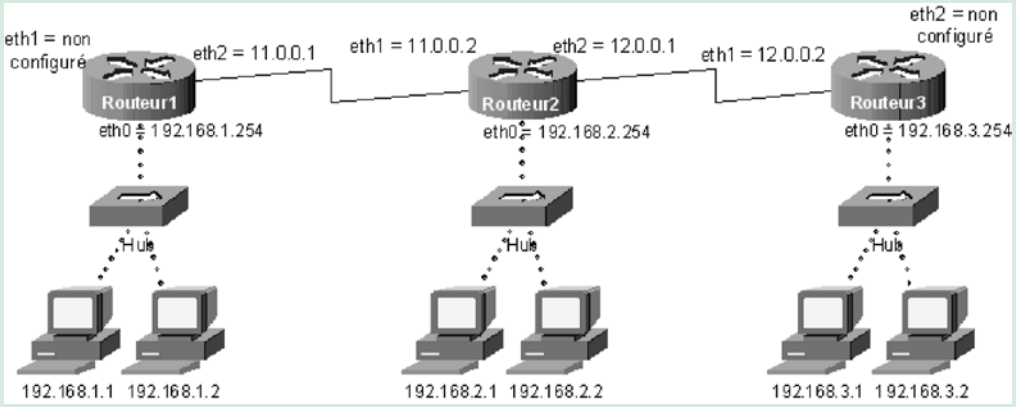


Figure 3 : topologie de réseau d'expérimentation

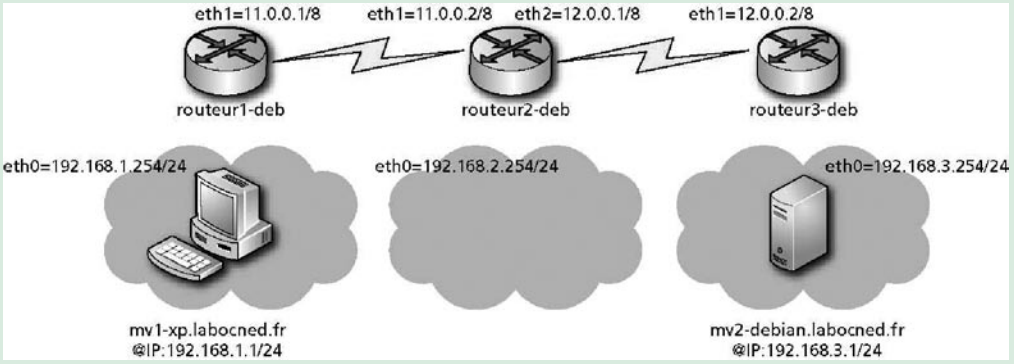


Schéma du réseau de l'atelier

Nous avons besoin de trois machines sous Linux qui joueront le rôle de routeur en disposant chacune de trois interfaces réseau. Sur une interface, un réseau local sera connecté. Les liens représentés par des éclairs symbolisent un réseau étendu. Ils seront simulés avec un câble croisé Ethernet (ou via un *hub* ou un *switch*) pour ceux qui feraient les ateliers avec du matériel physique.
Vous devez donc installer trois ordinateurs avec plusieurs cartes réseau dans chacun.

Mise en place du serveur : mv2-debian.labocned.fr

Il n'y a que l'adresse IP à retoucher à notre actuelle machine virtuelle. Vous verrez cela en temps utile lorsque vous ferez vos essais de connexions.

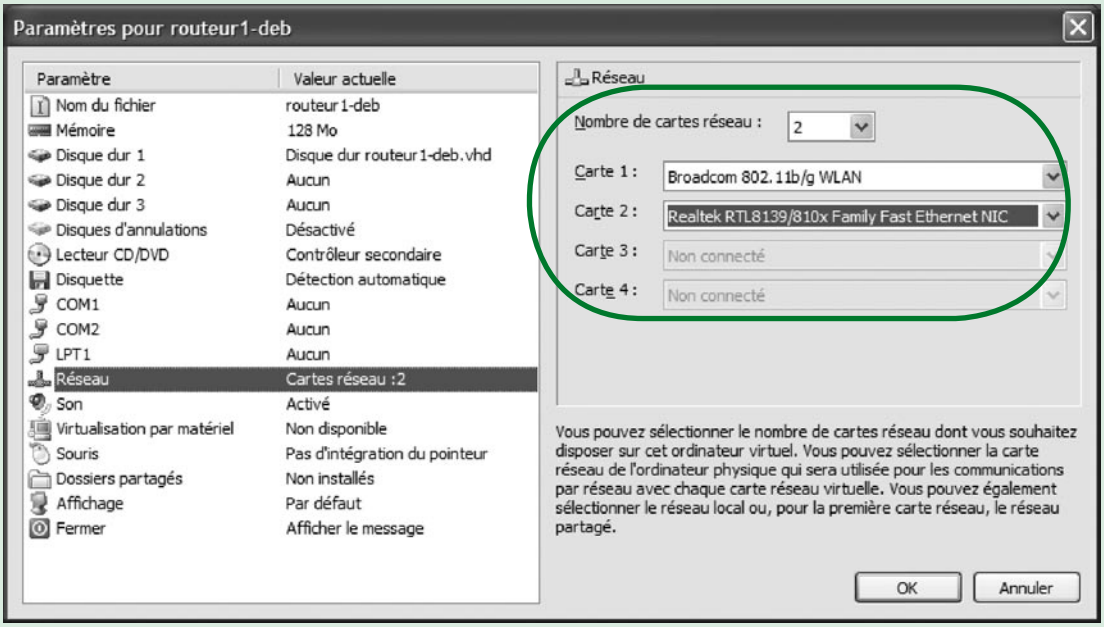
Mise en place du poste client : mv1-xp.labocned.fr

A priori il n'y a pas grand chose à faire dessus pour le moment. L'adresse IP sera changée au moment opportun, et, si le routage n'est pas fonctionnel nous savons qu'il ne pourra plus contacter son domaine et tous les services proposés par mv2-debian…

Mise en place des routeurs

Créons 3 machines virtuelles : routeur1-deb, routeur2-deb et routeur3-deb. Avec un disque dur virtuel pour chacune… Jusque là vous avez déjà fait…
Sélectionnons routeur1-deb par exemple (vous ferez de même pour les autres après…)

Et cliquez sur **Paramètres**. Sélectionnez le **Paramètre Réseau** pour configurer le nombre de cartes réseaux dont nous avons besoin pour chaque routeur.
Virtual PC permet d'émuler jusqu'à 4 cartes réseaux : pour chaque machine virtuelle indiquez le nombre de cartes réseaux à émuler, et sélectionnez l'interface physique correspondante.



Installez maintenant Debian sur chacun des routeurs. Lors de l'installation Debian vous demandera de choisir une interface réseau préférentielle (celle qui sera configurée pour pouvoir récupérer les paquets nécessaires sur Internet). Pour chacune vous laissez eth0 et configurez les adresses qui vous permettront de vous configurer sur Internet **sans vous préoccuper à ce moment-là de la configuration du schéma de l'atelier**.

Si vous avez un problème de dépendances, n'oubliez pas :

```
# apt-get –f install
```

Je vais prendre exemple sur mon routeur2-deb (car lui a 3 interfaces réseaux), après installation la configuration des interfaces est :

```
# ifconfig –a
eth0   […]
      inet adr :192.168.1.252 Bcast :192.168.1.255 Masque :255.255.255.0
      UP […]
eth1   […]
      Reconnue mais pas d'adresse, de masque…
eth2   idem
```

Donc elles ne sont pas configurées par rapport au schéma de réseau de l'atelier à réaliser, eth0 permet de se connecter à Internet.
Essayez : #apt-get install zebra. Vous obtiendrez un message comme quoi il n'a pas récupéré le paquet, mais qu'il en a trouvé trace. Il suggère que zebra est obsolète. Presque… Le paquet zebra n'est tout simplement plus maintenu, il est remplacé par Quagga qui intègre Zebra en tant que demon gestionnaire du routage.
Ensuite, installez le paquetage quagga : # apt-get install quagga
Configurez maintenant à l'aide de la commande ifconfig les diverses interfaces des routeurs, en exemple pour mon routeur2-deb cela donne :

```
# ifconfig eth0 192.168.2.254 netmask 255.255.255.0 up
```

Le up sert à activer l'interface… De même pour les autres interfaces :

```
# ifconfig eth1 11.0.0.2 netmask 255.0.0.0 up
eth1: Using EEPROM-set media 100baseTx-FDX.
# ifconfig eth2 12.0.0.1 netmask 255.0.0.0 up
eth2: Using EEPROM-set media 100baseTx-FDX.
```

Et vous verifiez tout cela :

```
# ifconfig –a
```

Configurez les interfaces des routeurs routeur1-deb et routeur3-deb conformes au schéma.

• Le routeur logiciel zebra/quagga

Tout d'abord, présentons le logiciel : comme vous avez pu le constater si vous avez essayé apt-get install zebra, zebra n'est plus maintenu et son successeur est quagga, mais cela ne change pas grand chose.
http://www.zebra.org/
http://www.quagga.net/

Architecture de Zebra

Zebra fonctionne sous Linux et BSD. C'est un routeur multi-protocole composé d'une suite de démons, un par protocole de routage dynamique plus un démon central (zebra) utilisé pour le routage statique. De plus, lorsque le routage dynamique est mis en œuvre, il est chargé de synthétiser dans une table de routage unique, les informations rapportées par les autres démons.

Son architecture générale est la suivante :

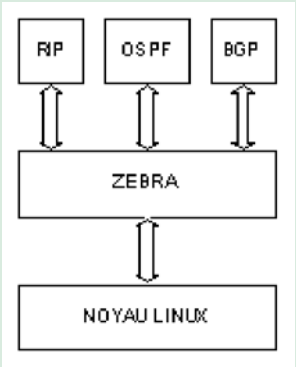


Figure 4 : architecture de Zebra

Démarrage du démon

Quagga, comme nos autres paquetages dans les ateliers précédents, possède son répertoire dans /etc :

```
# ls /etc/quagga
daemons  debian.conf
```

La documentation est incluse dans le paquetage et disponible sur le site dans divers formats (dont.pdf) http://www.quagga.net/docs.php. Lisez-la.

Quagga est composé de 5 démons (processus) en utilisation et un démon manager :

- ripd, ripngd, ospfd, ospf6d, bgpd (démons prenant en charge les protocoles de routage) ;
- zebra (le manager).

Chacun de ces démons écoute sur un port et peut être « accédé » par terminal.
Chacun de ces démons a également son fichier.conf : il faut donc créer le nôtre…

Zebra ne peut pas démarrer sans un fichier de configuration minimal qu'il faudra saisir sur chaque routeur. Éditez un fichier /etc/quagga/zebra.conf :

```
# vi /etc/quagga/zebra.conf
hostname Zebra
password routeurcned
```

Lorsque le service Quagga se lance, il exécute son fichier /etc/quagga/daemons pour savoir quel(s) démon(s) il doit démarrer, éditez-le pour obtenir :

```
zebra=yes
bgpd=no
ospfd=no
ospf6d=no
ripd=no
ripngd=no
isisd=no
```

Et relancez quagga :

```
# /etc/init.d/quagga restart
```

Connexion au routeur

Quagga dispose d'une interface telnet pour chaque démon. Pour configurer le démon zebra, il faut se connecter sur le port 2601. Par exemple, sur routeur2-deb :

```
# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.5).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Zebra>
```

À l'invite Password:, saisissez le mot de passe que vous avez indiqué dans le fichier /etc/quagga/zebra.conf (aucun caractère n'apparaît lors de la saisie, c'est « normal »). L'invite devient Zebra> (Zebra est le nom par défaut que vous changerez plus tard).

• Prise en main de Zebra (principes)

L'invite

L'invite (*prompt*) a une importance capitale. À la suite du nom de l'appareil, on peut savoir dans quel « méandre » de l'arborescence des menus de configuration on se trouve.
Une invite terminée par un > indique que vous êtes dans le mode VIEW : c'est le mode activé lorsque l'on se connecte. Comme son nom l'indique, il s'agit d'un mode de visualisation de l'état du routeur. Par exemple :

```
Zebra>
```

Une invite terminée par un # indique que vous êtes dans le mode ENABLE : c'est le mode privilégié qui permet de modifier la configuration du routeur. On y accède à partir du mode VIEW en tapant la commande enable. Par exemple :

```
Zebra> enable
Zebra#
```

Ensuite, vous travaillerez essentiellement avec deux sous-modes :
- le mode « terminal de configuration ». On y accède à partir du mode ENABLE en tapant la commande configure terminal. Dans ce mode, on peut (entre autre) saisir des routes statiques.

Par exemple :

```
Zebra# configure terminal
Zebra(config)#
```

- le mode « interface ». On y accède à partir du mode « terminal de configuration » en tapant la commande interface suivi du nom de l'interface réseau. Dans ce mode, on peut (entre autre) saisir l'adresse IP et le masque de l'interface.

Par exemple :

```
Zebra(config)# interface eth0
Zebra(config-if)#
```

L'aide en ligne

Retenez les éléments suivants, ils vous seront d'un grand secours :
- comme dans l'interpréteur de commandes Linux, le logiciel complète toutes les commandes lorsque vous appuyez sur la touche TAB (par exemple : shTAB devient show) ;
- un simple appui sur ? indique toutes les commandes disponibles dans le mode dans lequel vous êtes ;
- enfin, la commande list donne la liste de toutes les commandes disponibles dans le mode dans lequel vous êtes et de leurs paramètres.

• Prise en main de Zebra (mise en pratique)

Si vous n'êtes pas dans le mode VIEW, tapez end puis disable :

```
Zebra(config-if)# end
Zebra# disable
Zebra>
```

Utilisation de l'aide en ligne

Exercice 30

- Listez l'ensemble des commandes disponibles à ce stade.
- Listez toutes les commandes disponibles à ce stade avec tous leurs paramètres.
- Listez tous les paramètres de la commande qui permet d'afficher la configuration (commande show).

Affichage de l'état du routeur

Listez les interfaces disponibles sur votre routeur avec la commande show interface (n'oubliez pas qu'un simple shTAB intTAB suffit !) :

```
Zebra> show interface
Interface eth0 is up, line protocol detection is disabled
  index 2 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:03:ff:2e:42:02
  inet 192.168.1.252/24 broadcast 192.168.1.255
  inet6 fe80::203::ffff:fe2e:4202/64
    219 input packets (0 multicast), 242960 bytes, 0 dropped,
    0 input errors, 0 length, 0 overrun, 0 CRC, 0 frame
    0 fifo, 0 missed
    11 output packets, 762 bytes, 0 dropped
    0 output errors, 0 aborted, 0 carrier, 0 fifo, 0 heartbeat
    0 window, 0 collisions
Interface eth1 is down
  index 3 metric 1 mtu 1500
  flags <BROADCAST,MULTICAST>
  HWaddr: 00:03:ff:29:42:02
  [...]
Interface eth2 is down
  index 4 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:03:ff:2b:42:02
  [...]
```

Vous avez quatre interfaces (lo et trois Ethernet). Zebra a gardé la configuration IP au moment du dépaquetage de Quagga : les modifications (ifconfig) n'ont pas été prises en compte. Nous allons le faire.

Visionnez le contenu de votre table de routage : shTAB ip roTAB :

```
Zebra> show ip route
Codes:  K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
        I – ISIS, B - BGP, > - selected route, * - FIB route

K>* 0.0.0.0/0 via 192.168.1.1, eth0
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.1.0/24 is directly connected, eth0
```

La 1re route (K) correspond à la configuration de passerelle par défaut que nous avons effectué à l'installation. 127.0.0.0/8 correspond à l'adresse de bouclage logiciel.
Et enfin la route du réseau auquel appartient l'interface elle-même.
Pour le reste, le routeur n'a aucune idée de la façon d'acheminer les paquets de données vers tel ou tel réseau : il enverra vers la passerelle par défaut.
Le but de ce TP est de configurer correctement les interfaces et les tables de routage pour que l'ensemble des appareils puissent s'atteindre au travers des trois routeurs.

Configuration générale du routeur

• Passage au mode privilégié

Toute configuration nécessite de passer en mode privilégié :

```
Zebra> enable
Zebra#
```

Le # indique que vous êtes en mode privilégié.
Listez les commandes disponibles :

```
Zebra# ?
  configure  Configuration from vty interface
  copy       Copy configuration
  debug      Debugging functions (see also 'undebug')
  disable    Turn off privileged mode command
  echo       Echo a message back to the vty
  end        End current mode and change to enable mode.
  exit       Exit current mode and down to previous mode
  help       Description of the interactive help system
  list       Print command list
  logmsg     Send a message to enabled logging destinations
  no         Negate a command or set its defaults
  quit       Exit current mode and down to previous mode
  show       Show running system information
  terminal   Set terminal line parameters
  who        Display who is on vty
  write      Write running configuration to memory, network, or terminal
```

Les commandes importantes à ce stade sont **copy** pour enregistrer la configuration et **configure** pour accéder au terminal de configuration des interfaces réseau et des routes.

• Affichage de la configuration actuelle

La commande **write term** permet de consulter à tout moment la configuration actuelle du routeur :

```
Zebra# write terminal

Current configuration:
!
hostname Zebra
password routeurcned
!
interface eth0
  ipv6 nd suppress-ra
!
interface eth1
  ipv6 nd suppress-ra
!
interface eth2
  ipv6 nd suppress-ra
!
interface lo
!
interface sit0
  ipv6 nd suppress-ra
line vty
!
end
```

Rien n'est configuré, alors allons-y.

• Passage au mode « terminal de configuration »

```
Zebra# configure terminal
Zebra(config)#
```

• Définition du nom du routeur

Cela se fait par la commande hostname. Attribuez à chaque routeur un nom significatif. Par exemple, pour le routeur 2 :

```
Zebra(config)# hostname routeur2-deb(Zebra)
routeur2-deb(Zebra)(config)#
```

• Sécurisation du routeur

L'accès au routeur et à son mode de configuration doit être protégé.

• Activez le chiffrement des mots de passe.
Lorsque nous avons utilisé la commande write term, nous avons pu constater que le mot de passe n'était pas chiffré.

Résolvons ce problème :

```
routeur2-deb(Zebra)(config)# service password-encryption
```

• Attribuez un mot de passe pour l'accès au mode VIEW du routeur (en remplacement de « foo »). Il doit commencer par une lettre ou un chiffre :

```
routeur2-deb(Zebra)(config)# password ******
```

• Attribuez un mot de passe pour l'accès au mode ENABLE du routeur. Il doit également commencer par une lettre ou un chiffre :

```
routeur2-deb(Zebra)(config)# enable password ******
```

• Consultez la configuration :

```
Routeur2-deb(Zebra)(config)# write terminal

Current configuration:
!
hostname routeur2-deb(Zebra)
password 8 7cwtLNkwin4Qc
enable password 8 9Ghk/7S7yaDo6
service password-encryption
!
interface eth0
   ipv6 nd suppress-ra
!
interface eth1
   ipv6 nd suppress-ra
!
interface eth2
 ipv6 nd suppress-ra
!
interface lo
!
interface sit0
   ipv6 nd suppress-ra
line vty
!
end
```

Les mots de passe sont chiffrés maintenant.

• Enregistrement de la configuration

Il faut revenir au mode ENABLE. N'oubliez pas de sauvegarder régulièrement !

```
Routeur2-deb(Zebra)(config)# end
Routeur2-deb(Zebra)# copy running-config startup-config
Configuration saved to /etc/quagga/zebra.conf
```

Configuration des interfaces réseau

Il faut attribuer une adresse IP et un masque à chaque carte réseau. Pour cela, il faut se placer dans le mode « configuration d'interface » puis utiliser une commande ip address. Enfin, il faut activer l'interface avec une commande no shutdown.
Regardez le schéma réseau de l'atelier et exemple pour l'interface eth0 du routeur 2 :

```
Routeur2-deb(Zebra)# configure terminal
Routeur2-deb(Zebra)(config)# interface eth0
Routeur2-deb(Zebra)(config-if)# ip addr 192.168.2.254/24
Routeur2-deb(Zebra)(config-if)# no shutdown
```

Exercice 31

- Faites la même manipulation, en adaptant, pour les deux autres interfaces de ce routeur, puis affichez la configuration.
- Faites ensuite ces configurations sur les trois routeurs.

Configuration des routes

Allez dans le shell Linux du routeur2-deb et listez sa table de routage :

```
# route
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref  Use  Iface
192.168.2.0     *               255.255.255.0   U     0      0      0    eth0
localnet        *               255.255.255.0   U     0      0      0    eth0
11.0.0.0        *               255.0.0.0       U     0      0      0    eth1
12.0.0.0        *               255.0.0.0       U     0      0      0    eth2
default         192.168.1.1     0.0.0.0         UG    0      0      0    eth0
```

Cet appareil intègre automatiquement les adresses des réseaux directement connectés. Ainsi, Routeur2-deb peut communiquer avec l'interface de Routeur1-deb située dans le même réseau (11.0.0.0/8), et aussi avec l'interface de Routeur3-deb sur leur réseau commun (12.0.0.0/8).

```
# ping 12.0.0.2 -c 2
PING 12.0.0.2 (12.0.0.2) 56(84) bytes of data.
64 bytes from 12.0.0.2: icmp_seq=0 ttl=255 time=10.3 ms
64 bytes from 12.0.0.2: icmp_seq=1 ttl=255 time=1.55 ms

--- 12.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 1.553/5.931/10.309/4.378 ms
```

Tous les appareils peuvent pinger leurs voisins immédiats, mais pas au-delà. Par exemple, Routeur3 ne peut pas atteindre Routeur1 :

```
Routeur3-deb:# ping 11.0.0.1
PING 11.0.0.1 (11.0.0.1) 56(84) bytes of data.
From 192.168.3.254 icmp_seq=1 Destination Host Unreachable
From 192.168.3.254 icmp_seq=2 Destination Host Unreachable
```

Normal, puisque le réseau 11.0.0.0/8 ne figure pas dans sa table de routage !
Le routage statique va consister à intégrer manuellement les routes dans la table de routage du routeur. Ouvrez sur routeur3-deb une session telnet avec zebra puis allez dans le mode « configuration de terminal ».
La saisie des routes se fait avec la commande ip route. Son format général est :

```
ip route <adresse_reseau_destination> <adresse_ip_prochain_routeur>
```

Par exemple, sur Routeur3 pour atteindre Routeur1, il faudra indiquer 11.0.0.0/8 en réseau de destination et l'adresse IP de l'interface de Routeur2 qui est située du côté de Routeur3 :

```
Routeur3-deb(Zebra)(config)# ip route 11.0.0.0/8 12.0.0.1
```

Tentons à nouveau un ping dans le shell Linux :

```
Routeur3-deb(Zebra)(config)# end
Routeur3-deb(Zebra)# quit
Connection closed by foreign host.
Routeur3-deb# ping 11.0.0.1
PING 11.0.0.1 (11.0.0.1) from 12.0.0.2 : 56(84) bytes of data.

CTRL-C

--- 11.0.0.1 ping statistics ---
13 packets transmitted, 0 packets received, 100% packet loss
```

J'ai dû interrompre la commande car elle restait bloquée. Cela ne marche donc pas. Pourquoi ?
Notez tout d'abord la différence du message :

- précédemment nous avions formellement « From 192.168.3.254 icmp_seq=1 Destination Host Unreachable » : ne connaissant pas la route vers le réseau 11.0.0.0/8 le routeur3-deb essayait d'envoyer les paquets via sa passerelle par défaut et concluait que l'hôte n'était pas joignable ;
- maintenant ce message a disparu, le message est rien (=attente, de quoi ?…), on stoppe (CTRL C), et il conclue par des statistiques : 13 paquets émis, 0 reçus…

Pour comprendre, nous allons utiliser la commande Linux tcpdump❶ qui affiche tous les paquets passant par une interface. Sur Routeur3, ouvrez un autre terminal (ALT+F2) puis connectez-vous en root. Lancez la commande tcpdump -i eth1 puis revenez dans le premier terminal.

Relancez le ping précédent puis revenez dans le deuxième terminal et observez :

```
Routeur3-deb:# tcpdump -i eth1
tcpdump: listening on eth1
03:44:45.507835 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:44:46.507835 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:44:47.507835 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
```

Le routeur émet des paquets mais on constate qu'il n'y a pas de retour. Le problème peut venir de Routeur2 ou de Routeur1. Suivons la chaîne et connectons-nous dans le shell de Routeur2. Utilisons tcpdump sur l'interface du côté de Routeur3 :

```
Routeur2-deb# tcpdump -i eth2
tcpdump: listening on eth2
03:52:20.017855 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:52:21.017855 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:52:22.027855 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
```

Les paquets sont bien reçus. Faisons la même manipulation, toujours sur Routeur2 mais sur l'interface du côté de Routeur1 :

```
Routeur2-deb# tcpdump -i eth1
tcpdump: listening on eth1
03:52:46.027855 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:52:47.017855 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:52:48.017855 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
```

❶ *Installez le paquetage si nécessaire : apt-get install tcpdump (attention à la connexion internet…)*

Les paquets sont bien transmis d'une interface à l'autre. Routeur2 fait son travail. Passons sur Routeur1, faisons un tcpdump sur l'interface côté Routeur 2 :

```
Routeur1-deb# tcpdump -i eth1
tcpdump: listening on eth1
03:53:53.224868 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:53:54.224868 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:53:55.234868 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
```

Les paquets sont bien reçus sur Routeur1. Mais il n'est pas capable d'y répondre. D'où vient le problème alors ? Il n'y a qu'une solution. Il vient de la table de routage. Routeur1 ne sait pas comment répondre à Routeur3 car il n'a aucune idée de l'endroit où il se trouve puisqu'il n'est pas directement relié à lui. Connectons-nous au démon zebra sur Routeur1 puis visionnons sa table de routage :

```
Routeur1-deb(Zebra)> show ip route
Codes:  K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
        I – ISIS, B - BGP, > - selected route, * - FIB route

K>* 0.0.0.0/0 via 192.168.1.1, eth0
C>* 11.0.0.0/8 is directly connected, eth1
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.1.0/24 is directly connected, eth0
```

Elle ne contient aucune entrée pour 12.0.0.0/8, le réseau auquel est connecté Routeur3. Réparons cette erreur en informant Routeur1 sur la façon d'atteindre 12.0.0.0/8. Il faut passer par son voisin Routeur2 dont l'adresse est 11.0.0.2. Sur le routeur (en mode « terminal de configuration »), il faut saisir :

```
Routeur1-deb(Zebra)(config)# ip route 12.0.0.0/8 11.0.0.2
```

Affichons à nouveau la table de routage :

```
Routeur1-deb(Zebra)(config)# end
Routeur1-deb(Zebra)# show ip route
Codes:  K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
        I – ISIS, B - BGP, > - selected route, * - FIB route

K>* 0.0.0.0/0 via 192.168.1.1, eth0
C>* 11.0.0.0/8 is directly connected, eth1
S>* 12.0.0.0/8 [1/0] via 11.0.0.2, eth1
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.1.0/24 is directly connected, eth0
```

La route apparaît. Elle est notée S pour Statique.
Revenons dans le shell Linux de Routeur1-deb. La commande tcpdump nous montre que maintenant, Routeur1-deb sait répondre à Routeur3-deb :

```
Routeur1-deb:# tcpdump -i eth1
tcpdump: listening on eth1
03:59:51.904868 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:59:51.904868 11.0.0.1 > 12.0.0.2: icmp: echo reply
03:59:52.864868 12.0.0.2 > 11.0.0.1: icmp: echo request (DF)
03:59:52.864868 11.0.0.1 > 12.0.0.2: icmp: echo reply
```

Si l'on revient sur Routeur3, le ping fonctionne désormais sans problème :

```
Routeur3-deb:# ping 11.0.0.1 -c 2
PING 11.0.0.1 (11.0.0.1) from 12.0.0.2 : 56(84) bytes of data.
Warning: time of day goes back, taking countermeasures.
64 bytes from 11.0.0.1: icmp_seq=0 ttl=254 time=8.090 msec
64 bytes from 11.0.0.1: icmp_seq=1 ttl=254 time=3.986 msec

--- 11.0.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/mdev = 3.986/6.038/8.090/2.052 ms
```

Résumé

1. Le routage est une chaîne : les paquets sont transmis au routeur voisin, qui transmet au routeur voisin, etc.
2. Il faut penser à configurer tous les routeurs pour l'aller ET le retour des paquets.

Travaux pratiques

Exercice 32

Maintenant que vous savez tout, saisissez sur chaque routeur **l'ensemble minimal** des routes qu'il a besoin de connaître. Le cahier des charges est le suivant : les stations de chaque réseau (vous avez mv1-xp et mv2-debian qui, moyennant le changement de leur configuration IP, peuvent simuler client et serveur pour différents réseaux…) ont besoin de communiquer avec les stations de tous les autres réseaux. Les stations n'ont pas besoin de communiquer avec les routeurs.

D'abord, commencez par détruire les routes statiques que nous venons de saisir pour les besoins de la démonstration. Par exemple, sur Routeur3-deb pour détruire la route 11.0.0.0/8 :

```
Routeur3-deb(Zebra)> enable
Password:
Routeur3-deb(Zebra)# configure terminal
Routeur3-deb(Zebra)(config)# no ip route 11.0.0.0/8 12.0.0.1
Routeur3-deb(Zebra)(config)#
```

Dernier point : n'oubliez pas de configurer la passerelle par défaut sur les stations, sans quoi elles ne sauraient pas comment faire sortir leurs paquets de leur réseau local. La passerelle par défaut de chaque LAN est le routeur le plus proche. Par exemple sur une station du premier réseau local (192.168.1.0/24 : la passerelle sera 192.168.1.254), si c'est sous Windows :

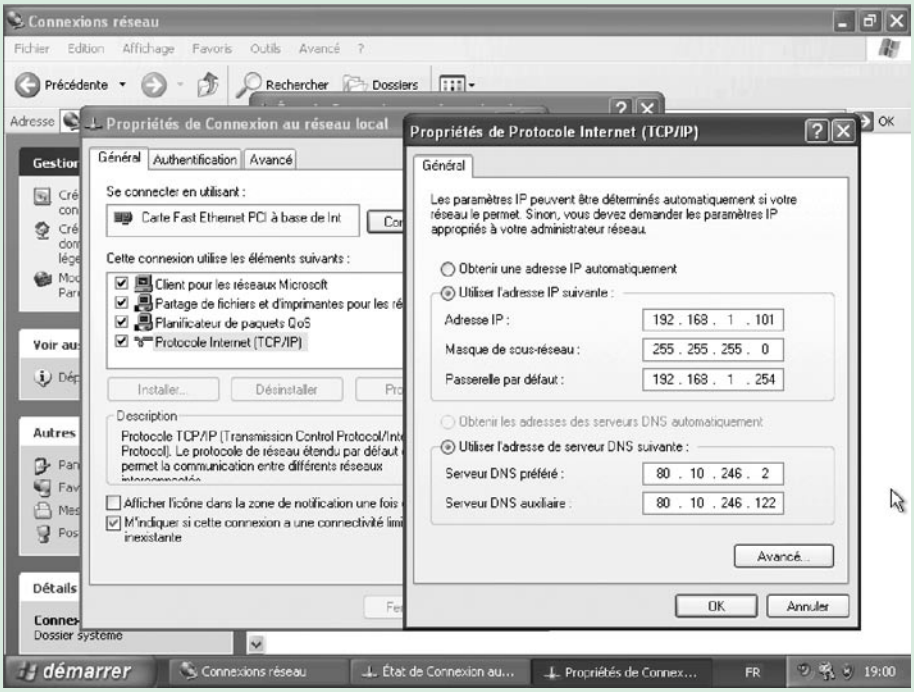

Connexions réseau
Propriétés de Connexion au réseau local
Propriétés de Protocole Internet (TCP/IP)
Général
Authentification
Avancé
Se connecter en utilisant :
Carte Fast Ethernet PCI à base de Int
Cette connexion utilise les éléments suivants :
Client pour les réseaux Microsoft
Partage de fichiers et d'imprimantes pour les ré
Planificateur de paquets QoS
Protocole Internet (TCP/IP)
Installer...
Désinstaller
Description
Protocole TCP/IP (Transmission Control Protocol/Int
Protocol). Le protocole de réseau étendu par défaut
permet la communication entre différents réseaux
Afficher l'icône dans la zone de notification une fois
M'indiquer si cette connexion a une connectivité lim
inexistante
Les paramètres IP peuvent être déterminés automatiquement si votre réseau le permet. Sinon, vous devez demander les paramètres IP appropriés à votre administrateur réseau.
Obtenir une adresse IP automatiquement
Utiliser l'adresse IP suivante :
Adresse IP :
192 . 168 . 1 . 101
Masque de sous-réseau :
255 . 255 . 255 . 0
Passerelle par défaut :
192 . 168 . 1 . 254
Obtenir les adresses des serveurs DNS automatiquement
Utiliser l'adresse de serveur DNS suivante :
Serveur DNS préféré :
80 . 10 . 246 . 2
Serveur DNS auxiliaire :
80 . 10 . 246 . 122
Avancé...
OK
Annuler
démarrer
Connexions réseau
État de Connexion au...
Propriétés de Connex...
FR
19:00

Figure 5 : configuration de la passerelle par défaut

Dépannages

Q - Je ne sais pas retirer les adresses IP déjà configurées sous Linux.

R - Éditez le contenu des fichiers /etc/network/interfaces et supprimez les lignes incriminées. Ensuite, redémarrez le service : /etc/init.d/networking restart

http://linux.developpez.com/faq/?page=configdebian#debnetconf

Q - Je n'arrive pas ouvrir de session Telnet.

R - Soit vous n'indiquez pas le numéro de port (2601) après l'adresse IP, soit le démon zebra n'est pas lancé. Faites un ps -ax pour vérifier. S'il n'apparaît pas, faites un zebra -d.

Q - J'ai oublié un mot de passe.

R - Bravo. Éditez le fichier /etc/quagga/zebra.conf et retirez les lignes password et service password-encryption.

Q - J'ai saisi une mauvaise commande et je ne sais pas la supprimer.

R - Vous devez faire exactement comme si vous vouliez saisir à nouveau cette commande mais vous faites précéder le tout de no. Exemple :

J'ai saisi une mauvaise adresse (99.0.0.9/8) sur l'interface eth0 de mon routeur :

```
Routeur1-deb(Zebra)> enable
Password:
Routeur1-deb(Zebra)# configure terminal
Routeur1-deb(Zebra)(config)# interface eth0
Routeur1-deb(Zebra)(config-if)# no ip address 99.0.0.9/8
Routeur1-deb(Zebra)(config-if)#
```

Résumé

Un routeur utilise une table de routage pour router les paquets qu'il reçoit. Router consiste à recevoir un paquet sur une interface et choisir sur quelle autre interface l'appareil doit le transmettre. Ce mécanisme est à la base d'internet.
Une table de routage est constituée des réseaux connus du routeur, du masque réseau associé, de la passerelle et de l'interface réseau à utiliser.
Le routage statique consiste à saisir manuellement les routes dans la table de routage.

Si vous voulez approfondir

Vous pouvez consulter le site : http://www.pmassol.net pour un début de traduction de la documentation de Zebra et des articles sur le sujet publiés dans Linux Magazine.
Le logiciel Zebra n'est plus actif, mais le site propose toujours des informations (http://www.zebra.org). Mais voyez plutôt du côté de Quagga (http://www.quagga.net).

Ces livres traitent du routage :

- COMER (Douglas), *TCP/IP : ARCHITECTURE, PROTOCOLE ET APPLICATIONS*, DUNOD. ISBN : 2-10-005384-1, 09/2001, p. 830
- HUITEMA (Christian), *LE ROUTAGE DANS L'INTERNET*, EyROLLES. ISBN : 2-212-08902-3, 10/1994, p. 418

Le routage IP dynamique OSPF

Durée approximative de cet atelier : 4 heures

▶ *Objectif*

Une fois la séquence finie, vous comprendrez le fonctionnement du protocole OSPF et vous saurez faire une configuration élémentaire du routage dynamique afin que vos routeurs s'échangent leurs informations.

▶ *Conditions préalables*

Idem atelier précédent.

▶ *Considérations techniques*

Idem atelier précédent.

▶ *Avertissement*

Idem atelier précédent.

Introduction

Ne faire que du routage statique sous Zebra n'a pas vraiment de sens. Le noyau de Linux sait déjà très bien le faire. Ce qu'il ne sait pas faire, c'est mettre en œuvre un protocole d'échange des informations de routage en sa possession. Ici, Zebra, va pleinement jouer son rôle.

• Avant OSPF

Historiquement, un des premiers protocoles de routage IP dynamique est RIP *(Routing Information Protocol)*. C'est un excellent moyen pédagogique pour aborder la problématique du routage dynamique. Mais, en réalité, il est très peu utilisé en exploitation car il souffre de certaines limitations et défauts qui le cantonnent à des réseaux de taille moyenne :

- le diamètre maximum d'un réseau géré avec RIP est limité à 15 routeurs soit 16 segments de réseau ;
- RIP est un gros consommateur de bande passante du fait de la méthode utilisée pour diffuser les informations de routage (toutes les 30 secondes, l'intégralité de la table RIP est diffusée même si elle n'a subi aucune modification). C'est fâcheux, en particulier sur des liaisons lentes ou facturées au volume de données transférées ;

- **la métrique**❶ utilisée (qui reflète le nombre de routeurs à traverser) ne garantit pas que le routage soit optimal. En effet, cette « distance » masque les caractéristiques réelles de la voie de transmission (débit ou coût en particulier).

• OSPF

Compte tenu de ces contraintes et afin de gagner du temps, j'ai choisi d'aborder directement OSPF qui est LE protocole de routage de prédilection à l'intérieur des grands réseaux d'entreprise.

Généralités

OSPF est un protocole de routage dynamique défini par l'IETF❷ à la fin des années 80. Il a fait l'objet d'un historique relativement complexe de RFC. Ce protocole a deux caractéristiques essentielles :

- il est ouvert (le *Open* de *OSPF*), son fonctionnement peut être connu de tous ;
- il utilise l'algorithme SPF (*Shortest Path First*), plus connu sous le nom d'algorithme de Dijkstra, afin d'élire la meilleure route vers une destination donnée.

Examinons une topologie qui nous servira de support pour les explications.

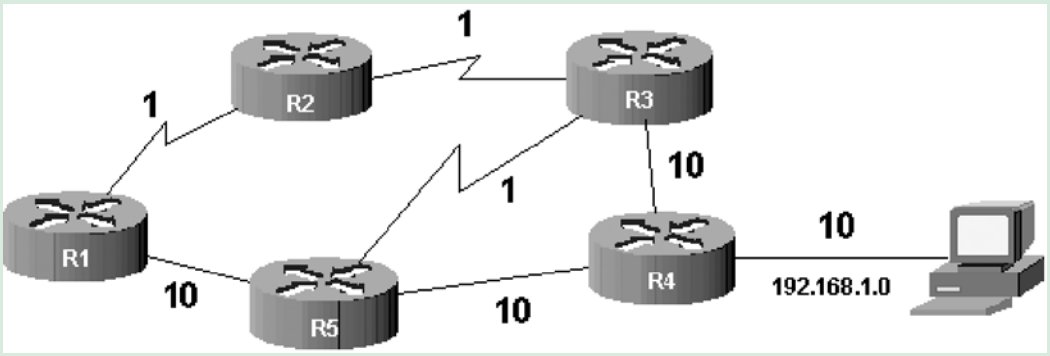


Figure 1 : un exemple de topologie

La notion de coût

Supposons que du routeur R1 on cherche à atteindre le réseau 192.168.1.0. Dans une telle situation, RIP aurait élu la route passant par R5 puisque c'est la plus courte en termes de saut❸. Cependant, imaginez que les liens représentés sous forme d'éclairs soient « rapides » (type Ethernet à 100 Mbps par exemple) et que les liens « droits » soient « lents » (type Ethernet à 10 Mbps par exemple). Le choix de RIP n'est plus du tout pertinent !

❶ *La métrique est un concept important dans le routage dynamique. C'est une information associée à une route que le routeur va évaluer afin de choisir la meilleure route vers une destination.*
❷ *Internet Engineering Task Force : organisme d'internet chargé de normaliser et de faire évoluer le réseau.*
❸ *Au nombre de routeurs traversés si vous préférez.*

OSPF fonctionne différemment. Il attribue un coût à chaque liaison (dénommée *lien* dans le jargon OSPF) afin de privilégier l'élection de certaines routes. Plus le coût est faible, plus le lien est intéressant. Par défaut, les coûts suivants sont utilisés en fonction de la bande passante du lien :

Type de réseau	Coût par défaut
Ethernet > = 100 Mbps	1
FDDI	1
Ethernet 10 Mbps	10
E1 (2,048 Mbps)	48
T1 (1,544 Mbps)	65
64 Kbps	1562
56 Kbps	1785
19.2 Kbps	5208

La formule de calcul est simplissime : coût = référence / bande passante du lien. Par défaut, la référence est 100 000 000 correspondant à un réseau à 100 Mbps.

OSPF privilégie les routes qui ont un coût faible, donc celles qui sont supposées rapides en terme de débit théorique.

La base de données topologique

Avec OSPF, tous les routeurs d'un même réseau (on parle de « zone » dans le vocabulaire OSPF, ceci vous sera expliqué avant la mise en pratique) travaillent sur une base de données topologique identique qui décrit le réseau. Cette base a été constituée pendant une première phase de découverte qui vous sera expliquée un peu plus loin. Examinons la base de données suivante qui décrit la topologie de la figure 1 :

Arc	Coût
R1, R2	1
R1, R5	10
R2, R3	1
R3, R4	10
R3, R5	1
R4, R5	10
R4, 192.168.1.0	10

L'élection des meilleures routes

L'algorithme du SPF de Dijsktra va traiter cette base de données afin de déterminer les routes les moins coûteuses. Une fois le traitement réalisé, chaque routeur se voit comme la racine d'un arbre contenant les meilleures routes.

Par exemple :

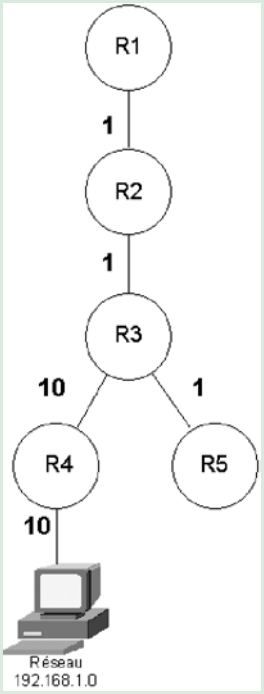


Figure 2 : le réseau vu de R1

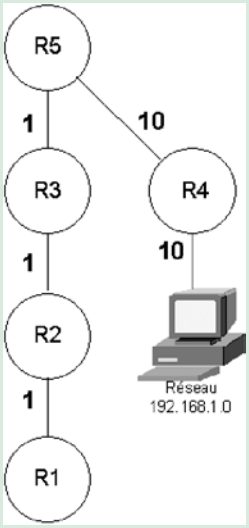


Figure 3 : le réseau vu de R5

Dans l'exemple, entre R1 et 192.168.1.0, la meilleure route passe par R2, R3 et R4 pour un coût total de 1 + 1 + 10 + 10 soit 22.

La détermination d'une table de routage

La base de données topologique décrit le réseau mais ne sert pas directement au routage. La table de routage est déterminée par l'application de l'algorithme du SPF sur la base topologique. Sur R1, voici un extrait de la table de routage calculée par SPF au sujet du réseau 192.168.1.0 :

Réseau de destination	Moyen de l'atteindre	Coût
192.168.1.0	R2	22

Sur R5, on aura l'extrait suivant :

Réseau de destination	Moyen de l'atteindre	Coût
192.168.1.0	R4	20

Le fonctionnement d'OSPF un peu plus en détail

Pour administrer un réseau OSPF correctement, il est indispensable de comprendre le fonctionnement interne du protocole.
À l'intérieur d'une même zone, les routeurs fonctionnant sous OSPF doivent préalablement remplir les tâches suivantes avant de pouvoir effectuer leur travail de routage :

1. établir la liste des routeurs voisins ;
2. élire le routeur désigné (et le routeur désigné de secours) ;
3. découvrir les routes ;
4. élire les routes à utiliser ;
5. maintenir la base de donnée topologique.

0. État initial

Le processus de routage OSPF est inactif sur tous les routeurs de la figure 1.

1. Établir la liste des routeurs voisins : *Hello, my name is R1 and I'm an OSPF router*

Les routeurs OSPF sont bien élevés. Dès qu'ils sont activés, ils n'ont qu'une hâte : se présenter et faire connaissance avec leurs voisins. En effet, lorsque le processus de routage est lancé sur R1 (commande `router ospf`), des paquets de données (appelés paquets HELLO) sont envoyés sur chaque interface où le routage dynamique a été activé (commande `network`). L'adresse *multicast* 224.0.0.5 est utilisée, tout routeur OSPF se considère comme destinataire. Ces paquets ont pour but de s'annoncer auprès de ses voisins. Deux routeurs sont dits voisins s'ils ont au moins un lien en commun. Par exemple, sur la figure 1, R1 et R2 sont voisins mais pas R1 et R3.
Lorsque le processus de routage OSPF est lancé sur R2, celui-ci récupère les paquets HELLO émis par R1 toutes les 10 secondes (valeur par défaut du temporisateur appelé *hello interval*). R2 intègre l'adresse IP de R1 dans une base de données appelée « base d'adjacences » (*adjacencies database*). Cette base contient les adresses des routeurs voisins. Vous pourrez visionner son contenu grâce à la commande `show ip ospf neighbor`. R2 répond à R1 par un paquet IP *unicast*. R1 intègre l'adresse IP de R2 dans sa propre base d'adjacences. Ensuite, généralisez ce processus à l'ensemble des routeurs de la zone.
Cette phase de découverte des voisins est fondamentale puisque OSPF est un protocole à état de liens. Il lui faut connaître ses voisins pour déterminer s'ils sont toujours joignables et donc déterminer l'état du lien qui les relie.

2. Élire le routeur désigné : c'est moi le chef !

Dans une zone OSPF, l'un des routeurs doit être élu « routeur désigné » (DR pour *Designated Router*) et un autre « routeur désigné de secours » (BDR pour *Backup Designated Router*). Le DR est un routeur particulier qui sert de référent au sujet de la base de données topologique représentant le réseau.

Pourquoi élire un routeur désigné ? Cela répond à trois objectifs :

- réduire le trafic lié à l'échange d'informations sur l'état des liens (car il n'y a pas d'échange entre tous les routeurs mais entre chaque routeur et le DR) ;
- améliorer l'intégrité de la base de données topologique (car il y a une base de données unique) ;
- accélérer la convergence en cas de modification de la topologie du réseau (panne d'un routeur par exemple).

Comment élire le DR ? Autrement dit, qui va se taper la corvée d'expliquer à ses petits camarades la topologie du réseau ? On ne demande pas qui sait parler anglais ou couper les cheveux comme au temps de la conscription. Mais comme il faut bien un critère, le routeur élu est celui qui a la plus grande priorité. La priorité est un nombre sur 8 bits fixé par défaut à 1 sur tous les routeurs. Pour départager les routeurs ayant la même priorité, c'est celui avec la plus grande adresse IP qui est élu. Le BDR sera le routeur avec la deuxième plus grande priorité. Afin de s'assurer que votre routeur préféré sera élu DR, il suffit de lui affecter une priorité supérieure à 1 avec la commande ospf priority. Vous devrez faire ceci avant d'activer le processus de routage sur les routeurs car, une fois élu, le DR n'est jamais remis en cause même si un routeur avec une priorité plus grande apparaît dans la zone.

3. Découvrir les routes

Il faut maintenant constituer la base de données topologique. Les routeurs communiquent automatiquement les routes pour les réseaux qui participent au routage dynamique (ceux déclarés avec la commande network). Quagga/Zebra étant multiprotocole, il peut également diffuser des routes provenant d'autres sources que OSPF, grâce à la commande redistribute.

Chaque routeur (non DR ou BDR) établit une relation maître/esclave avec le DR. Le DR initie l'échange en transmettant au routeur un résumé de sa base de données topologique via des paquets de données appelés LSA (*Link State Advertisement*). Ces paquets comprennent essentiellement l'adresse du routeur, le coût du lien et un numéro de séquence. Ce numéro est un moyen pour déterminer l'ancienneté des informations reçues. Si les LSA reçus sont plus récents que ceux dans sa base topologique, le routeur demande une information plus complète par un paquet LSR (*Link State Request*). Le DR répond par des paquets LSU (*Link State Update*) contenant l'intégralité de l'information demandée. Ensuite, le routeur (non DR ou BDR) transmet les routes meilleures ou inconnues du DR.

L'administrateur peut consulter la base de données topologique grâce à la commande show ip ospf database.

4. Élire les routes à utiliser

Lorsque le routeur est en possession de la base de données topologique, il est en mesure de créer la table de routage. L'algorithme du SPF est appliqué sur la base topologique. Il en ressort une table de routage contenant les routes les moins coûteuses.

Il faut noter que sur une base de données topologique importante, le calcul consomme pas mal de ressources CPU car l'algorithme est relativement complexe.

5. Maintenir la base topologique

Lorsqu'un routeur détecte un changement de l'état d'un lien (cette détection se fait grâce aux paquets HELLO adressés périodiquement par le routeur à ses voisins), celui-ci émet un paquet LSU sur l'adresse *multicast* 224.0.0.6 : le DR et le BDR de la zone se considèrent comme destinataires. Le DR (et le BDR) intègre cette information à sa base topologique et diffuse l'information sur l'adresse 224.0.0.5 (tous les routeurs OSPF sans distinction). C'est le protocole d'inondation.

Toute modification de la topologie déclenche une nouvelle exécution de l'algorithme du SPF et une nouvelle table de routage est constituée.

Voilà pour les principes fondamentaux d'OSPF mais des notions importantes restent à évoquer si vous souhaitez déployer OSPF sur de grands réseaux (en particulier sur le fonctionnement d'OSPF sur un réseau point à point et sur l'agrégation de routes). Si vous voulez approfondir, reportez-vous au livre de C. Huitema cité en bibliographie qui, bien qu'un peu ancien est très complet sur la question. Bien sûr, vous pouvez toujours vous plonger dans les différentes RFC qui constituent OSPF (la RFC 2328 en particulier) et dont la lecture est toujours aussi agréable et passionnante ! (je plaisante, bien sûr).

Avant d'attaquer la pratique, un dernier concept : les zones OSPF.

Le concept de zone *(area)*

Contrairement à RIP, OSPF a été pensé pour supporter de très grands réseaux. Mais, qui dit grand réseau, dit nombreuses routes. Donc, afin d'éviter que la bande passante ne soit engloutie dans la diffusion des routes, OSPF introduit le concept de zone (*area*). Le réseau est divisé en plusieurs zones de routage qui contiennent des routeurs et des hôtes. Chaque zone, identifiée par un numéro, possède sa propre topologie et ne connaît pas la topologie des autres zones. Chaque routeur d'une zone donnée ne connaît que les routeurs de sa propre zone ainsi que la façon d'atteindre une zone particulière, la zone numéro 0. Toutes les zones doivent être connectées physiquement à la zone 0 (appelée *backbone* ou réseau fédérateur). Elle est constituée de plusieurs routeurs interconnectés. Le *backbone* est chargé de diffuser les informations de routage qu'il reçoit d'une zone aux autres zones. Tout routage basé sur OSPF doit posséder une zone 0.

Figure 4 : un réseau découpé en trois zones

Sur la figure 4, le réseau est découpé en trois zones dont le *backbone*. Les routeurs de la zone 1, par exemple, ne connaissent pas les routeurs de la zone 2 et encore moins la topologie de la zone 2. L'intérêt de définir des zones est de limiter le trafic de routage, de réduire la fréquence des calculs du plus court chemin par l'algorithme SPF ainsi que d'avoir une table de routage plus petite (ce qui accélère la convergence). Les routeurs R1 et R4 sont particuliers puisqu'ils sont « à cheval » entre plusieurs zones (on les appelle ABR pour *Area Border Router* ou routeur de bordure de zone). Ces routeurs maintiennent une base de données topologique pour chaque zone à laquelle ils sont connectés. Les ABR sont des points de sortie pour les zones ce qui signifie que les informations de routage destinées aux autres zones doivent passer par l'ABR local à la zone. L'ABR se charge alors de retransmettre les informations de routage au backbone. Les ABR du *backbone* ensuite redistribueront ces informations aux autres zones auxquelles ils sont connectés.

Place à la pratique

Nous allons travailler avec la topologie ci-dessous. Comme dans l'atelier précédent, pour être vraiment intéressant, un TP sur le routage nécessite plusieurs machines. Virtual PC (il existe d'autres logiciels de ce type, comme VMware) permet d'émuler toutes ces machines.

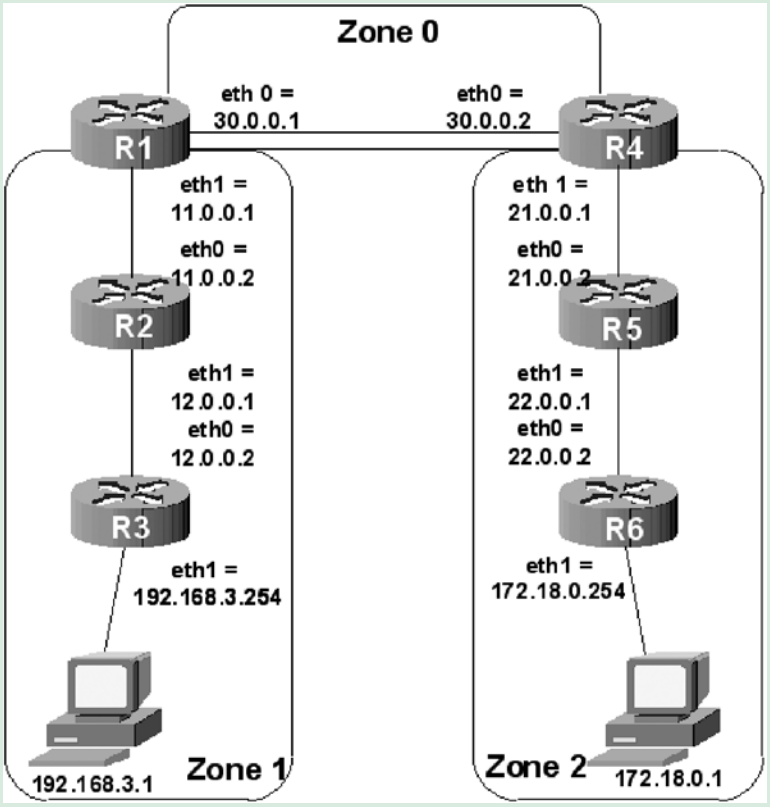


Figure 5 : la topologie de travail

Le réseau a été découpé en trois zones. Vous remarquez que la zone 0 permet de fédérer l'ensemble du réseau. Il s'agit du *backbone* dont nous avons déjà discuté. Le découpage de ce réseau en trois zones est un cas d'école dont le but est d'examiner la configuration d'OSPF dans un contexte multi-zone. Généralement, on considère qu'une zone peut accueillir plusieurs dizaines de routeurs.

Pour ne pas surcharger ces lignes inutilement, nous nous en tiendrons ici à la configuration de R1, R2 et R3. Vous verrez que la configuration n'est pas très complexe. Par symétrie, il est facile de l'adapter aux autres routeurs.

0. Situation de départ

Dans le shell de R1, assurez-vous que la configuration des interfaces est conforme au schéma de réseau proposé :

```
# cat /etc/network/interfaces
[…]
# Interface réseau eth0
auto eth0
iface eth0 inet static
        address 30.0.0.1
        netmask 255.0.0.0
        network 30.0.0.0
        broadcast 30.255.255.255
# Interface réseau eth1
auto eth1
iface eth1
        address 11.0.0.1
        netmask 255.0.0.0
        network 11.0.0.0
        broadcast 11.255.255.255
```

Vous devez créer des fichiers de configuration pour zebra❶ (/etc/quagga/zebra.conf) et ospfd (/etc/quagga/ospfd.conf) rudimentaires sur chaque routeur. Par exemple, pour R1 :

- fichier zebra.conf : (remplacez l'ancien s'il existe)

```
hostname routeur1-deb(ZEBRA)
password routeurcned
```

- fichier ospfd.conf :

```
hostname routeur1-deb(OSPF)
password routeurcned
```

ospfd acquiert les informations d'interface de zebra : c'est pour ça qu'avec la nouvelle configuration réseau nous avons réinitialisé /etc/quagga/zebra.conf et que vous devez réinitialiser les interfaces réseaux dans zebra.
Vous devez ensuite activer, les deux démons zebra et ospfd sur chaque routeur.

❶ *En principe, vous l'avez fait lors de l'atelier précédent.*

```
# vi /etc/quagga/daemons
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
# /etc/init.d/quagga restart
```

Enfin, sur R1 entrez dans le terminal de configuration de ospfd via le port telnet 2604 :

```
# telnet localhost 2604

Hello, this is Quagga (version 0.99.5).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Routeur1-deb(OSPF)> enable
Routeur1-deb(OSPF)#
```

Si vous avez envie de suivre précisément les échanges de messages entre routeurs, Zebra propose un puissant mécanisme de débogage grâce à la commande **debug** (je vous laisse découvrir tous ses paramètres). Supposons que je veuille garder une trace de tous les messages HELLO émis par R1 :

```
Routeur1-deb(OSPF)# configure terminal
Routeur1-deb(OSPF)(config)# log file /var/log/quagga/ospfd.log
Routeur1-deb(OSPF)(config)# debug ospf packet hello send detail
```

Vérifiez que le répertoire /var/log/quagga existe.

• Activation du processus de routage

Dans le mode « config », nous allons activer le processus OSPF :

```
Routeur1-deb(OSPF)(config)# router ospf
Routeur1-deb(OSPF)(config-router)#
```

• Activation des annonces de routes

Le processus de routage OSPF est activé mais rien ne se passe. Il faut indiquer sur quel(s) réseau(x) on souhaite que le routage dynamique soit opérationnel. Ceci se fait par la commande network. De plus, il faut indiquer à quelle zone sera rattaché le réseau. Sur la figure 5, on voit que R1 est relié à deux réseaux. Le réseau 30.0.0.0/8 est attaché à la zone 0 et le réseau 11.0.0.0/8 à la zone 1. La configuration se fait donc de cette manière :

```
Routeur1-deb(OSPF)(config-router)# network 30.0.0.0/8 area 0
Routeur1-deb(OSPF)(config-router)# network 11.0.0.0/8 area 1
```

Que se passe-t-il sur le réseau ? R1 envoie des paquets **HELLO** sur les interfaces pour lesquelles la commande network a été saisie. Mais personne n'est là pour les écouter. Activez le routage sur R2 en adaptant les commandes aux spécificités du routeur. Je vous aide un peu. Sur R2, vous réaliserez les configurations suivantes :

```
Routeur2-deb(OSPF)(config-router)# network 11.0.0.0/8 area 1
Routeur2-deb(OSPF)(config-router)# network 12.0.0.0/8 area 1
```

Enfin, sur R3, vous réaliserez les configurations suivantes :

```
Routeur3-deb(OSPF)(config-router)# network 12.0.0.0/8 area 1
Routeur3-deb(OSPF)(config-router)# network 192.168.3.0/24 area 1
```

Mais sur R3, il y a une particularité. Le réseau 192.168.3.0/24 contient des ordinateurs mais aucun routeur. La commande network va diffuser sur ce réseau des annonces de routes ce qui consomme inutilement de la bande passante. Par conséquent, nous allons désactiver cette diffusion :

```
Routeur3-deb(OSPF)(config-router)# passive-interface eth1
```

Ainsi, aucune route n'est diffusée sur cette interface. De même, aucune annonce de route ne sera prise en compte. Le réseau sera considéré comme étant d'extrémité (*stub*).

Affichage de la configuration

Affichons la configuration complète de R1 :

```
Routeur1-deb(OSPF)(config-router)# end
Routeur1-deb(OSPF)# show running-config

Current configuration:
!
hostname routeur1-deb(OSPF)
password routeurcned
log file /var/log/quagga/ospfd.log
!
debug ospf packet hello send detail
!
!
interface eth0
!
interface eth1
!
interface lo
!
router ospf
network 11.0.0.0/8 area 0.0.0.1
network 30.0.0.0/8 area 0.0.0.0
!
line vty
!
end
```

Affichons la configuration complète de R3 :

```
Routeur3-deb(OSPF)# show running-config

Current configuration:
!
hostname routeur3-deb(OSPF)
password routeurcned
log file /var/log/quagga/ospfd.log
!
debug ospf packet hello send detail
!
!
interface eth0
!
interface eth1
!
interface lo
!
router ospf
passive-interface eth1
network 12.0.0.0/8 area 0.0.0.1
network 192.168.3.0/24 area 0.0.0.1
!
line vty
!
end
```

J'espère que vous avez la même configuration. Si ce n'est pas le cas, vous pouvez annuler une ligne contenant une erreur en vous remettant au même endroit où vous avez saisi la commande et en saisissant à nouveau la commande, mais en la faisant précéder de no. Pour enregistrer la configuration, je vous rappelle que l'on saisit :

```
Routeur1-deb(OSPF)# copy running-config startup-config
```

État des routeurs

Nos petits routeurs ont, en principe, bien travaillé. Dans chaque zone, ils ont élu leur chef (le DR), ils ont échangé leurs connaissances et calculé une magnifique table de routage, ultra-optimale. En résumé, les deux stations d'extrémité de la figure doivent pouvoir s'atteindre avec une commande ping. Si jamais ce n'est pas le cas, c'est que probablement vous vous êtes trompé dans une configuration. Dans ce cas, reprenez la configuration de chaque appareil. Utilisez les outils ping, tcpdump et traceroute pour contrôler votre configuration et suivre les paquets. Et n'oubliez pas que dans un ping, il y a un aller mais aussi un retour !
Afin d'illustrer ce dont nous avons discuté dans la toute première partie de cet article, examinons l'état du routeur R1. Nous pouvons faire un diagnostic très complet de l'appareil en utilisant les nombreuses sous-commandes de show ip ospf. Vous constaterez que les informations fournies par ospfd sur son état sont très détaillées.

Dans un premier temps, je vous propose d'examiner l'état de santé général du routeur R3 :

```
Routeur3-deb(OSPF)# show ip ospf
  OSPF Routing Process, Router ID: 192.168.3.254
  Supports only single TOS (TOS0) routes
  This implementation conforms to RFC2328
  RFC1583Compatibility flag is disabled
  OpaqueCapability flag is disabled
  Initial SPF schedule delay 200 millisec(s)
  Minimum hold time between consecutive SPFs 1000 millisec(s)
  Maximum hold time between consecutive SPFs 10000 millisec(s)
  Hold time multiplier is currently 2
  SPF algorithm last executed 3m10s ago
SPF timer is inactive
  Refresh timer 10 secs
  Number of external LSA 0. Checksum Sum 0x00000000
  Number of opaque AS LSA 0. Checksum Sum 0x00000000
  Number of areas attached to this router: 1

  Area ID: 0.0.0.1
    Shortcutting mode: Default, S-bit consensus: ok
    Number of interfaces in this area: Total: 2, Active: 2
    Number of fully adjacent neighbors in this area: 1
    Area has no authentication
    Number of full virtual adjacencies going through this area: 0
    SPF algorithm executed 13 times
    Number of LSA 9
```

Le premier bloc décrit le fonctionnement général du routeur : l'ID du routeur (égale à sa plus grande adresse IP), conformité aux RFC, valeurs des temporisateurs. Une seule zone est attachée à ce routeur. C'est la zone 1 (exprimée en notation décimale pointée). Notre routeur a deux interfaces dans la zone, il n'a qu'un seul voisin. L'algorithme du SPF a été exécuté 13 fois. La base de données topologique contient neuf états de liens (LSA). Si notre routeur était attaché à plusieurs zones, le deuxième bloc serait répété autant de fois que de zones. Vous pourrez le constater sur R1.

Maintenant, listons nos informations sur les routeurs voisins :

```
Routeur3-deb(OSPF)# show ip ospf neighbor
Neighbor  ID  PRI State          Dead  Time  AddressInter  RXmtL     RqstL DbsmL
                                                      face
12.0.0.1      1   Full/Backup 00:00:34  12.0.0.1 eth0    0         0     0
```

Déchiffrons ces informations. La différence entre la colonne ID et la colonne *Address*, c'est que l'ID identifie l'appareil dans le réseau alors que l'adresse correspond à l'interface à laquelle nous sommes reliés avec ce routeur. La colonne *State* nous apprend deux choses : il est synchronisé avec le routeur désigné grâce à la mention « *Full* », c'est le « routeur désigné de secours » de la zone grâce à l'indicateur *Backup*. Ce routeur sera déclaré comme inactif si nous ne recevons pas de message HELLO d'ici 34 secondes (*Dead Time*).
Voyons le contenu de la base de données topologique de R3 :

```
Routeur3-deb(OSPF)# show ip ospf database
            OSPF Router with ID (192.168.3.254)
                                Router Link States (Area 0.0.0.1)
Link ID         ADV Router      Age  Seq#       CkSum  Link count
12.0.0.1        12.0.0.1        981  0x80000006 0xf9e2 2
30.0.0.1        30.0.0.1        952  0x80000003 0xb13e 1
192.168.3.254   192.168.3.254   1063 0x80000005 0x15b7 2

                                Net Link States (Area 0.0.0.1)
Link ID         ADV Router      Age  Seq#       CkSum
11.0.0.2        12.0.0.1        981  0x80000001 0xda39
12.0.0.2        192.168.3.254   1063 0x80000001 0x5d0b

                            Summary Link States (Area 0.0.0.1)
Link ID         ADV Router      Age  Seq#       CkSum  Route
21.0.0.0        30.0.0.1        837  0x80000001 0x0d08 21.0.0.0/8
22.0.0.0        30.0.0.1        702  0x80000001 0x64a5 22.0.0.0/8
30.0.0.0        30.0.0.1        976  0x80000001 0x33e2 30.0.0.0/8
172.18.0.0      30.0.0.1        599  0x80000001 0x4a0d 172.18.0.0/24
```

Ces trois tableaux présentent de façon synthétique l'ensemble des LSA stockés dans la base topologique. Le premier tableau contient des LSA diffusés par chaque routeur. Ils décrivent l'état des interfaces de chaque routeur. Le deuxième tableau contient des LSA diffusés par le routeur désigné. Ils décrivent la liste des routeurs présents dans chaque réseaux. Le dernier tableau contient un résumé des routes diffusées par le routeur de bordure de zone (ABR). Ce sont des routes qu'il a reçu via le *backbone* par les routeurs des autres zones. L'âge et le numéro de séquence sont utilisés pour mettre à jour la base lorsque des LSA sont reçus. Le *check sum* est utilisé pour contrôler l'intégrité des LSA.
Pour obtenir des informations détaillées sur chaque LSA, vous pouvez compléter la commande show ip ospf database par router, network ou summary. Par exemple : show ip ospf database router 192.168.3.254 (qui correspond à la troisième ligne du premier tableau) vous apprendra que ce routeur est relié à deux réseaux : un de transit (12.0.0.0/8) et un d'extrémité (*stub*) 192.168.3.0/24.
Enfin, si vous voulez consulter la table de routage obtenue après traitement par SPF des différents LSA, vous n'aurez qu'à saisir un show ip ospf route. Rappel important : il y a une différence entre cette table et celle utilisée par le démon zebra pour le routage proprement dit. Souvenez-vous que Quagga est multi-protocole et qu'il a une architecture modulaire. Chaque démon calcule une table de routage à partir des informations dont il dispose (et qui ne sont pas nécessairement les mêmes pour chaque démon). Ensuite, ils transmettent chacun leur table au démon zebra qui en fait la synthèse. Cette synthèse constitue la véritable table de routage utilisée pour router les paquets.
Nous avons fait un tour d'horizon des principales commandes de Quagga permettant de surveiller l'état de ospfd. Il y en a encore beaucoup d'autres que je vous laisse découvrir (faites un show ip ospf ? par exemple). Il nous reste à observer la table de routage obtenue par Quagga. Quittez ospfd et connectez-vous sur le démon zebra (telnet localhost 2601) :

```
Routeur3-deb(Zebra)> show ip route :    K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
      B - BGP, > - selected route, * - FIB route


O>* 11.0.0.0/8 [110/20] via 12.0.0.1, eth0, 00:12:48
O> 12.0.0.0/8 [110/10] is directly connected, eth0, 00:14:09
C>* 12.0.0.0/8 is directly connected, eth0
O>* 21.0.0.0/8 [110/40] via 12.0.0.1, eth0, 00:11:18
O>* 22.0.0.0/8 [110/50] via 12.0.0.1, eth0, 00:10:16
O>* 30.0.0.0/8 [110/30] via 12.0.0.1, eth0, 00:12:13
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.18.0.0/24 [110/60] via 12.0.0.1, eth0, 00:08:48
O> 192.168.3.0/24 [110/10] is directly connected, eth1, 00:14:19
C>* 192.168.3.0/24 is directly connected, eth1
```

Les routes notées O ont été découvertes par OSPF. Entre crochets, on observe la distance administrative du protocole (110 par défaut pour OSPF) et le coût de la route pour accéder au réseau. Dans ma topologie, il n'y a que des réseaux à 10 Mbits/s, donc avec un coût par défaut de 10 pour chaque lien.

Résumé

OSPF est un protocole de routage dynamique moderne, robuste et conçu pour les grands réseaux. On constate qu'il est nettement plus complexe que RIP. Pas forcément dans sa configuration mais dans son fonctionnement interne. Un inconvénient de ce protocole est qu'il peut être gourmand en puissance de calcul et en mémoire lorsque le réseau comporte beaucoup de routes ou qu'il y a de fréquentes modifications de topologie.

OSPF est un protocole IGP (*Interior Gateway Protocol*), c'est-à-dire qu'il agit au sein d'un système autonome. Un AS (*Autonomous System*) est un ensemble de réseaux gérés par un administrateur commun. Chaque système autonome possède un numéro identifiant sur 16 bits délivré par l'IANA (*Internet Assigned Numbers Authority*) ou ses délégations. Classiquement, les multinationales, les opérateurs de télécom ou les fournisseurs d'accès à Internet détiennent un système autonome. Pour assurer le routage entre les systèmes autonomes, un protocole de type EGP (*Exterior Gateway Protocol*) doit être mis en œuvre. Dans le cas d'Internet, c'est généralement BGP (*Border Gateway Protocol*) qui assume cette mission. BGP est un protocole supporté par Zebra.

OSPF est le protocole de routage dynamique le plus répandu à l'intérieur des réseaux d'entreprise. Les routeurs s'échangent des informations sur les réseaux qu'ils connaissent en affectant en particulier une métrique (un poids) à chaque réseau. À partir ce ces informations, l'algorithme du SPF qui tourne sur chaque routeur est capable de fabriquer une véritable carte topologique du réseau. Cette carte est optimisée puisque l'on cherche à établir les « meilleures » routes (meilleur peut vouloir dire « plus rapide », mais aussi « plus sûr » ou « moins coûteux »).

Si vous voulez approfondir

Il faudrait approfondir la question du filtrage de la diffusion des routes. En effet, le premier inconvénient d'un protocole de routage dynamique comme OSPF est sa volubilité. Il a tendance à dévoiler tout un tas d'informations sur les réseaux qu'un administrateur consciencieux n'a pas forcément envie de révéler. Pour limiter la diffusion des routes au strict minimum, ospfd intègre un mécanisme d'access-lists.

Le deuxième inconvénient d'un protocole de routage dynamique comme OSPF est sa naïveté. Il croit tout ce qu'on lui dit ! Un petit malin pourrait s'amuser à diffuser des routes farfelues à vos routeurs, ce qui pourrait provoquer des dénis de service. Pour pallier cela, on peut activer l'authentification des annonces sur une zone grâce à une commande du style `area 1 authentication message-digest`.

Audit de sécurité

Durée approximative de cet atelier : 4 heures

▶ *Objectif*

À la fin de cette séance, vous serez en mesure d'installer et mettre en œuvre le logiciel libre Nessus dont l'objectif est de dresser un bilan complet des éventuelles failles de sécurité d'un système informatique.

▶ *Considérations techniques*

Vous pouvez faire l'audit de n'importe quelle machine sous Linux ou windows. En ce qui concerne cet atelier, nous réaliserons l'audit de notre serveur Linux mv2-debian avec lequel nous travaillons depuis le début de ces ateliers. Elle est intéressante car nous avons installé de nombreux services réseau.
Nous utiliserons la version 3 de Nessus.

▶ *Téléchargement et documentation*

http://www.nessus.org

Introduction

Un système informatique est par définition composé de nombreux éléments (le noyau, les différentes librairies, les services, les applications...). Chacun est potentiellement une faille de sécurité. La difficulté pour l'administrateur consiste à suivre précisément toutes les « alertes » qui surviennent ici ou là sur internet et à mettre à jour régulièrement son système.
Un outil capable de tester automatiquement la solidité d'un système en recherchant dans une base de données à jour toutes les failles connues à un moment donné est un allié très précieux. C'est la raison pour laquelle, nous allons travailler avec le produit Nessus.

Présentation de Nessus

La source principale d'informations et de documentation est bien sûr le site officiel : http://www.nessus.org

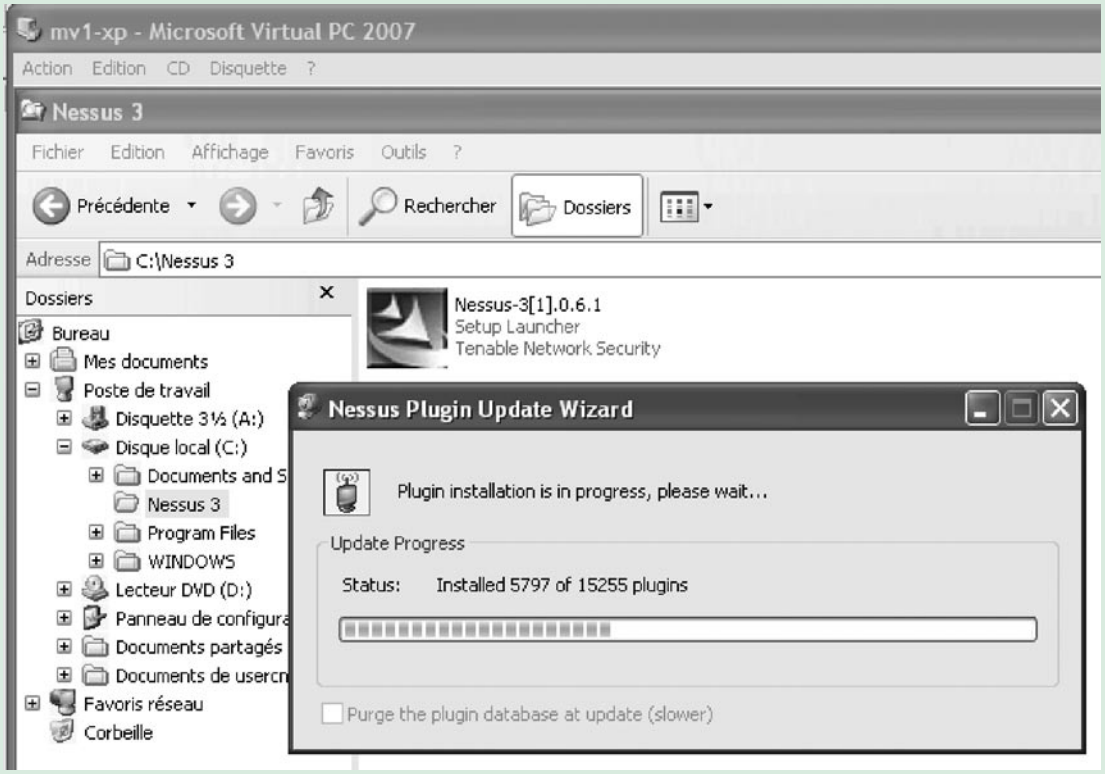
Mais pour avoir une présentation générale et synthétique je vous invite à consulter : http://fr.wikipedia.org/wiki/Nessus_%28logiciel%29

Nessus est un outil de sécurité informatique. Il signale les faiblesses potentielles ou avérées sur les machines testées. Ceci inclut, entre autres :

- les services vulnérables à des attaques permettant la prise de contrôle de la machine, l'accès à des informations sensibles (lecture de fichiers confidentiels par exemple), des dénis de service ;
- les fautes de configuration (relais de messagerie ouvert par exemple) ;
- les patchs de sécurité non appliqués, que les failles corrigées soient exploitables ou non dans la configuration testée ;
- les mots de passe par défaut, quelques mots de passe communs, et l'absence de mots de passe sur certains comptes systèmes. Nessus peut aussi appeler le programme externe **Hydra** pour attaquer les mots de passe à l'aide d'un **dictionnaire** ;
- les services jugés *faibles* (on suggère par exemple de remplacer Telnet par SSH) ;
- les dénis de service contre la pile TCP/IP.

Installation sur windows xp (mv1-xp)

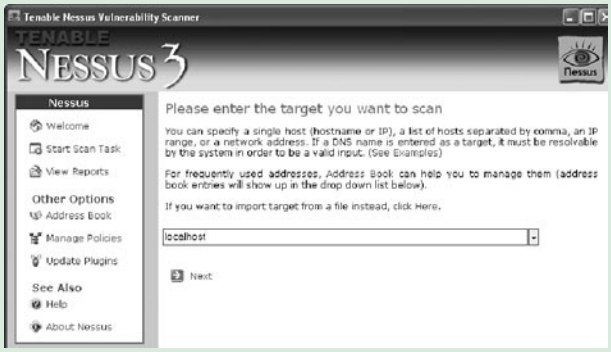
Nessus est composé d'une partie serveur et d'une partie client.
Sous mv1-xp téléchargez Nessus sur le site officiel, et inscrivez-vous : vous recevrez un code d'activation par e-mail.
Installez Nessus 3 sur mv1-xp, activez-le et téléchargez les mises à jour et plugins.



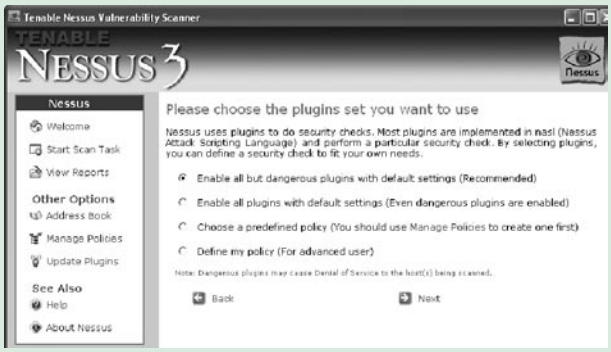
Lancez Nessus

Tenable Nessus Vulnerability Scanner
TENABLE
NESSUS 3
Nessus
Welcome
Start Scan Task
View Reports
Other Options
Address Book
Manage Policies
Update Plugins
See Also
Help
About Nessus
Welcome to Nessus Vulnerability Scanner
Nessus is a complete network vulnerability scanner which includes high-speed checks for thousands of the most commonly updated vulnerabilities, a wide variety of scanning options, an easy-to-use interface, and effective reporting.
You can start a new scan by selecting "Start Scan Task". All scan results will be automatically saved, and you can open them again by selecting "View Reports".
Start Scan Task
View Reports

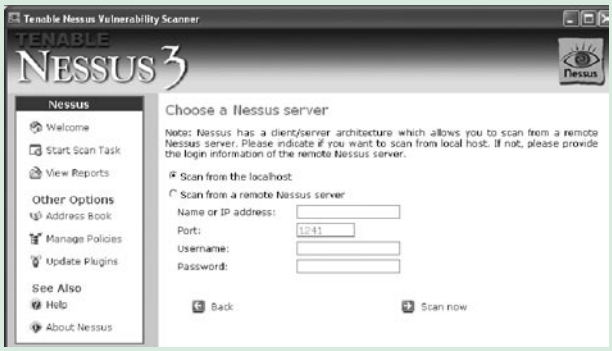
Écran d'accueil de Nessus
Choisissez Start Scan Task : nous allons tout d'abord effectuer un scan local sur cette machine avant de l'utiliser en réseau.


Tenable Nessus Vulnerability Scanner
TENABLE
NESSUS 3
Nessus
Welcome
Start Scan Task
View Reports
Other Options
Address Book
Manage Policies
Update Plugins
See Also
Help
About Nessus
Please enter the target you want to scan
You can specify a single host (hostname or IP), a list of hosts separated by comma, an IP range, or a network address. If a DNS name is entered as a target, it must be resolvable by the system in order to be a valid input. (See Examples)
For frequently used addresses, Address Book can help you to manage them (address book entries will show up in the drop down list below).
If you want to import target from a file instead, click Here.
localhost
Next

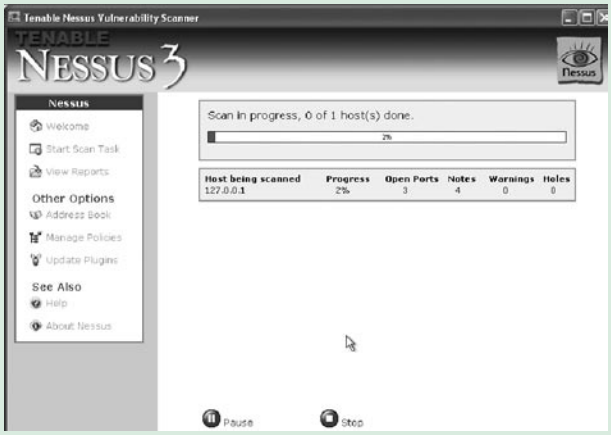
Sélectionnez dans la liste la seule donnée présente : localhost.
C'est ici qu'il faudra plus tard saisir l'adresse IP de l'hôte à scanner.
Cliquez Next.


Tenable Nessus Vulnerability Scanner
TENABLE
NESSUS 3
Nessus
Welcome
Start Scan Task
View Reports
Other Options
Address Book
Manage Policies
Update Plugins
See Also
Help
About Nessus
Please choose the plugins set you want to use
Nessus uses plugins to do security checks. Most plugins are implemented in nasl (Nessus Attack Scripting Language) and perform a particular security check. By selecting plugins, you can define a security check to fit your own needs.
Enable all but dangerous plugins with default settings (Recommended)
Enable all plugins with default settings (Even dangerous plugins are enabled)
Choose a predefined policy (You should use Manage Policies to create one first)
Define my policy (For advanced user)
Back
Next

Les plugins de Nessus sont en fait des programmes qui testent une ou plusieurs vulnérabilités : on peut donc personnaliser le type d'analyse que l'on veut effectuer en sélectionnant les plugins à utiliser.
Pour le moment voyons de quoi il retourne en laissant le choix proposé : enable all but dangerous… Cliquez Next.

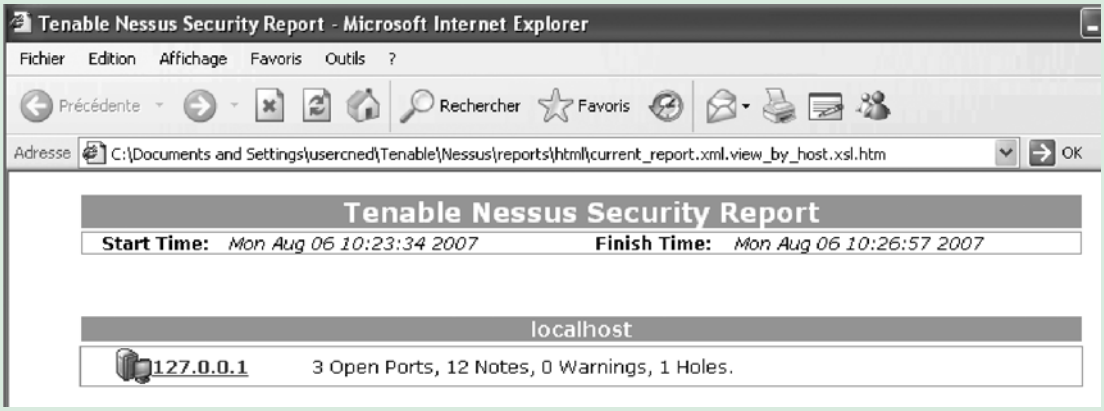


Nessus a une architecture client/serveur : il est donc intéressant en réseau d'installer un serveur dont le rôle sera d'inspecter et de surveiller la sécurité des hôtes et du réseau.
Pour notre premier scan nous sommes en localhost. Cliquez Scan now..



Après initialisation, le scan démarre.

Les résultats du scan sont disponibles au format.html :

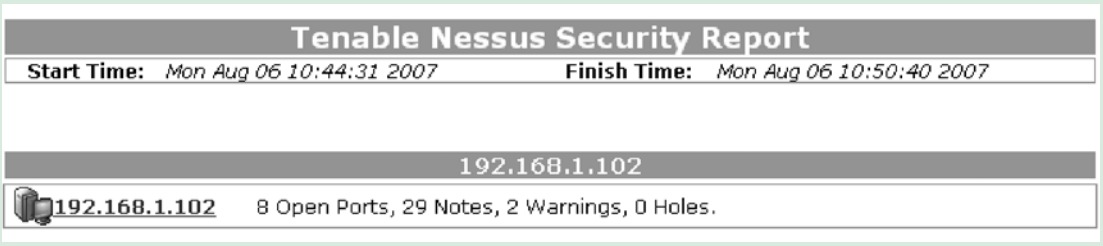


Interprétation des résultats
Nessus effectue un scan des ports (TCP et UDP) ouverts et teste les possibilités d'attaque par d'éventuelles failles de sécurité.
Le rapport indique ici 3 ports ouverts : 12 Notes et surtout 1 faille de sécurité : en lisant la suite du rapport nous avons le détail.
Par exemple la faille de sécurité de mon système est la possibilité d'exécuter du code par un hôte distant avec les privilèges système ! Mais Nessus m'indique aussitôt une solution : un patch sécurité Microsoft…

Scan de sécurité sur le réseau : scan du serveur debian

Relancez un scan depuis mv1-xp en indiquant en cible l'adresse IP du serveur debian mv2 : 192.168.1.102

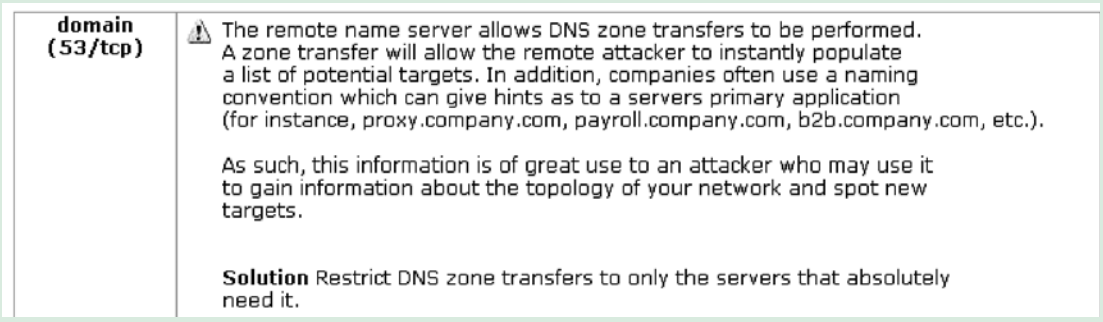
Interprétation des résultats


Tenable Nessus Security Report
Start Time: Mon Aug 06 10:44:31 2007
Finish Time: Mon Aug 06 10:50:40 2007
192.168.1.102
192.168.1.102
8 Open Ports, 29 Notes, 2 Warnings, 0 Holes.

8 ports ouverts ce n'est pas surprenant : depuis le début des ateliers nous avons installé de nombreux services sur cette machine.
La synthèse est :

- 29 notes (dans le rapport ils sont symbolisés par l'icône)
- 2 alertes (warnings, dont le symbole est :)
- 0 trous de sécurités (holes).

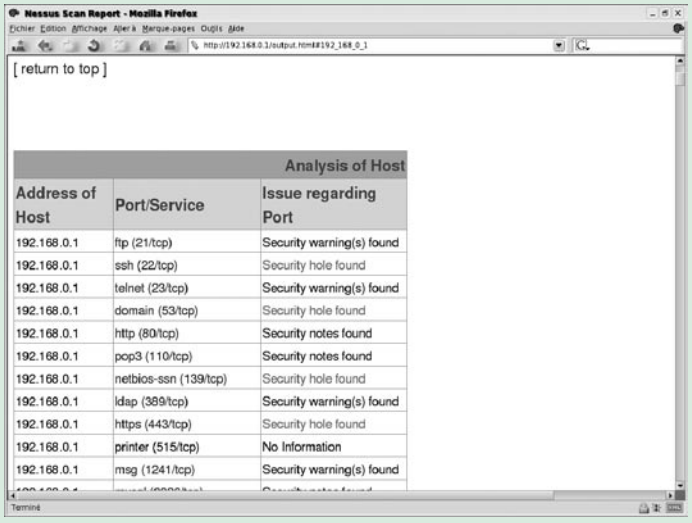
Je vous laisse interpréter vos « Notes », je vais commenter ici mes « Warnings » :


domain
(53/tcp)
The remote name server allows DNS zone transfers to be performed.
A zone transfer will allow the remote attacker to instantly populate
a list of potential targets. In addition, companies often use a naming
convention which can give hints as to a servers primary application
(for instance, proxy.company.com, payroll.company.com, b2b.company.com, etc.).
As such, this information is of great use to an attacker who may use it
to gain information about the topology of your network and spot new
targets.
Solution Restrict DNS zone transfers to only the servers that absolutely
need it.

Le 1er risque vient d'une possibilité de transfert de zone DNS : si une machine distante peut récupérer par transfert mes informations de zone DNS alors (surtout si mon domaine porte un nom explicite comme masociete.fr) il pourra reconstituer ma topologie réseau et identifier des cibles… Nessus me préconise en solution de restreindre les tranferts de zone DNS aux seuls serveurs qui en ont besoin (les serveurs DNS de mon réseau donc).

domain (53/udp)	**Synopsis** : The remote name server allows recursive queries to be performed by the host running nessusd. **Description** : It is possible to query the remote name server for third party names. If this is your internal nameserver, then forget this warning. If you are probing a remote nameserver, then it allows anyone to use it to resolve third parties names (such as www.nessus.org). This allows hackers to do cache poisoning attacks against this nameserver. If the host allows these recursive queries via UDP, then the host can be used to 'bounce' Denial of Service attacks against another network or system.

Le 2e warning concerne également l'administration de mon domaine (port UDP 53).
Ici Nessus me signale que mon serveur autorise les requêtes DNS distantes : ne pas m'inquiéter si c'est un serveur interne, mais s'il est externe cela veut peut engendrer une attaque par déni de service en effectuant des requêtes DNS récursives… et mon service tombe !
Bien qu'alarmant cela pourrait être pire :


Nessus Scan Report - Mozilla Firefox
[return to top]
Analysis of Host
Address of Host
Port/Service
Issue regarding Port
192.168.0.1 ftp (21/tcp) Security warning(s) found
192.168.0.1 ssh (22/tcp) Security hole found
192.168.0.1 telnet (23/tcp) Security warning(s) found
192.168.0.1 domain (53/tcp) Security hole found
192.168.0.1 http (80/tcp) Security notes found
192.168.0.1 pop3 (110/tcp) Security notes found
192.168.0.1 netbios-ssn (139/tcp) Security hole found
192.168.0.1 ldap (389/tcp) Security warning(s) found
192.168.0.1 https (443/tcp) Security hole found
192.168.0.1 printer (515/tcp) No Information
192.168.0.1 msg (1241/tcp) Security warning(s) found

Conclusion

Nessus est un outil performant qui peut éviter de parcourir sans cesse les forums, s'abonner à de multiples revues, des listes de diffusions, etc. Ceci dit, comme toujours dans le domaine de la sécurité, vous devez avoir une confiance limitée.

La plupart des problèmes de sécurité proviennent de services qui ne sont pas mis à jour ou mal configurés.

Sans tomber dans la paranoïa quelques recommandations :

- bien se documenter lorsque l'on installe et configure un service ;
- effectuer des mises à jour systématiques en étant abonné à des sources de sécurité par exemple : http://www.certa.ssi.gouv.Fr ;
- effectuer périodiquement des analyses.

Résumé

Nessus peut aider l'administrateur dans sa fastidieuse tâche de mise à jour de ses ordinateurs. Nessus est une base de données de (toutes ?) failles informatiques couplée à un logiciel simulant des attaques. Le résultat est un bilan indiquant toutes les sources potentielles de problème et la façon d'y remédier.

Si vous voulez approfondir

Lisez l'excellent magazine MISC dédié à la sécurité.

Séquence 6

NAS et SAN

Durée indicative : 1 heure

Cette séquence présente deux technologies de stockage proches l'une de l'autre.

▶ *Prérequis*

- **Aucun**

▶ *Contenu*

1. Introduction 181

2. NAS : Network Attached Storage 182

3. SAN : Storage Area Network 183

4. NAS ou SAN ? 184

▶ *Capacités attendues*

- **Être capable d'expliquer le principe de ces deux technologies.**
- **Être capable d'expliquer leur différence et d'évoquer leur convergence.**

1. Introduction

Dans certaines entreprises la capacité de stockage nécessaire peut devenir très importante. Peu importe que vos serveurs disposent de 5 disques durs SCSI de 160 Go chacun : nous parlons ici de besoins exprimés en To !!!… Ou plus !!!

C'est-à-dire qu'il n'est plus possible d'envisager les solutions de stockage dites en attachement direct (directement connecté) : **DAS *Direct Attached System***.

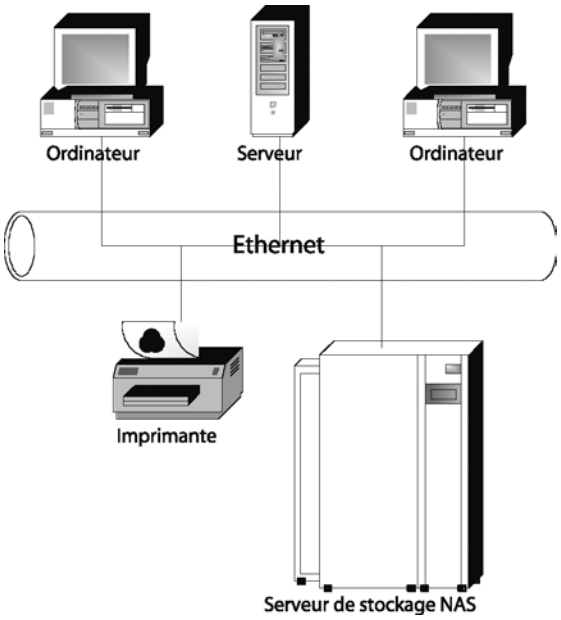
Dans ce cas là il faut alors avoir recours à d'autres solutions. Deux technologies dont les acronymes sont trompeurs sont envisageables :

- NAS : Network Attached Storage ;
- SAN : Storage Area Network.

Attention surtout à ne pas les confondre, mais cette séquence est là pour ça…

2. NAS : Network Attached Storage

Il est intéressant de commencer à définir la technologie NAS en disant qu'il s'agit d'un serveur dédié au stockage.
Le serveur dispose d'une adresse IP comme les autres… Les données qu'il stocke sont accessibles comme les autres…



Mais alors qu'a-t-il de plus qu'un serveur avec de nombreux disques ? :

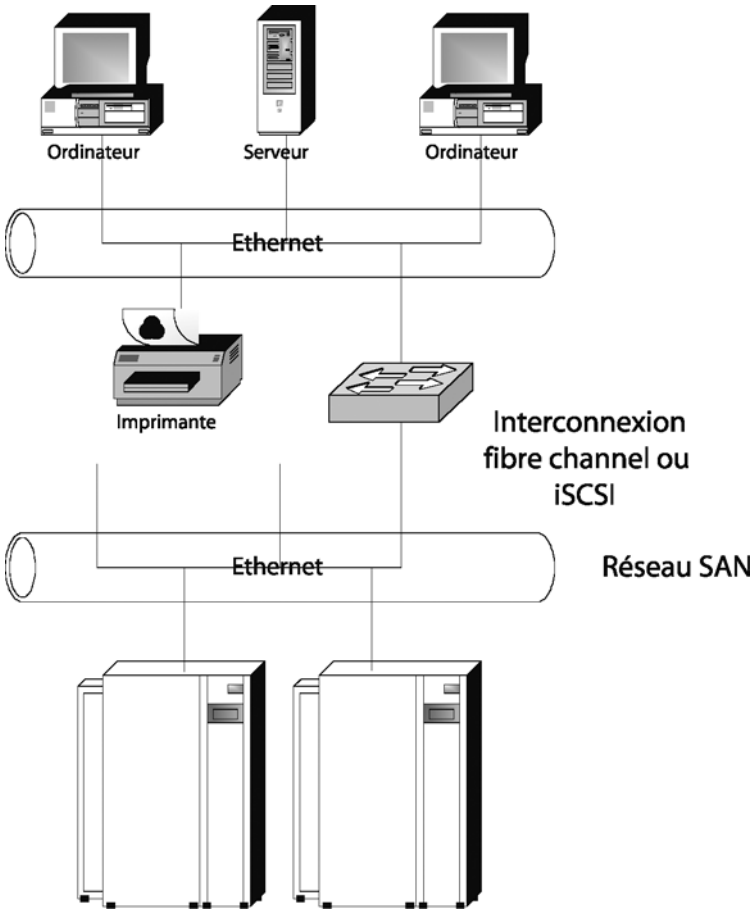
– son coût est moindre pour l'équivalent ;
– ses fonctionnalités sont intégrées : RAID 1 ou 5, sauvegarde intégrée… ;
– sa facilité d'intégration et de maintenance au sein de réseaux Ethernet.

C'est pour cela que des entreprises optent pour cette solution de serveurs pour des capacités de stockage atteignant les 100 To.

3. SAN : Storage Area Network

Même si l'acronyme est proche, jusqu'à constituer la source d'erreur la plus fréquente chez les étudiants, la simple traduction du sigle porte en elle la différence : réseau d'espace de stockage !

Alors que le ou les serveurs NAS sont directement connectés au réseau de l'entreprise, le SAN est un réseau à part : un réseau de stockage qui est interconnecté au réseau de l'entreprise.



Oh : fibre channel, iSCSI ??? aucun cours n'en a parlé je sais ; mais il n'est pas souhaitable d'écrire une encyclopédie, cela vous perdrait… Sachez simplement qu'il s'agit de deux technologies de connexion très rapides et que vous pourrez trouver davantage d'informations sur Internet.

4. NAS ou SAN ?

Vous avez bien compris que ces deux technologies proposent aux entreprises qui en ont besoin des solutions de stockage haute capacité et performance. C'est leur point commun.

Leurs différences :

- le NAS est directement accessible sur le LAN de l'entreprise ;
- le SAN est un réseau à part entière qui est interconnecté au LAN de l'entreprise.

Le SAN propose donc de bien meilleures caractéristiques… mais il est aussi beaucoup plus cher ! Chacune répond à un besoin différent et doit être adaptée au cas par cas.

Mais, pour vous donner une image, en général, le SAN conviendra peut-être mieux aux grandes entreprises tandis que le NAS répondra peut-être au besoin des PME-PMI.

Comparatif des deux technologies

- Qualité de service (Qos) : le LAN qui accueille le NAS n'offre pas les mêmes garanties que le SAN dont le switch assure en outre une garantie de débit (100 Mbits/s via fibre optique). Les entreprises qui ont des applications critiques nécessitant une haute qualité de service auraient intérêt à opter pour un SAN.
- Disponibilité : le SAN assure une redondance (accessibilité en cas de panne d'un de ses éléments) au niveau des cartes HBA (Host Bus Adapter), des serveurs et des switchs. Le NAS n'intègre pas nativement cette redondance.
- Hétérogénéité : le NAS s'adapte très bien aux environnements hétérogènes, il est interopérable avec des environnements UNIX et windows. L'interopérabilité d'un SAN est plus complexe étant donné le nombre de matériels (serveurs, switchs, baies de disques) issus de fabricants différents.
- Coût : solution plus simple le NAS n'a même pas besoin par exemple de liaison fibre spécialisée puisqu'il se connecte directement au LAN. Le SAN est évidemment beaucoup plus cher : mise en place d'un réseau spécialisé relié par fibre + de nombreux et onéreux matériels.

Convergence des deux technologies

Attention nous sommes ici en pleine anticipation : le sujet est d'actualité, les fabricants proposent leurs premières solutions… et je ne sais pas lire dans les lignes de la main. Prenez ces informations avec toutes les précautions d'usage, tenez une veille technologique sur le sujet.

Il y a c'est clair un vrai besoin de solutions intermédiaires : les deux technologies apparaissant comme extrêmes, soit pas chères mais pas top, soit excellentes mais très chères…

ISCSI est présenté comme une cause de cette convergence. ISCSI (Internet Small Computer System Interface) est un dérivé de notre SCSI pour les disques durs adapté aux réseaux TCP/IP. ISCSI peut donc remplacer la liaison fibre optique des SAN au détriment de la performance.

De plus les protocoles FCIP, iFCP ou encore infiniband pourraient apporter des réponses aux interconnexions de SAN ou NAS.

Bref ces deux technologies sont appelées à se développer vers un compromis entre haute disponibilité et coût. Et nous en resterons là pour le moment.